DELPHI'S

ORACLE

ORACLE

User's Manual

Copyright 1983 by Batteries Included

Program written by Dieter Demmer

1) All rights reserved. This manual is protected by copyright. No part of this manual may be reproduced in any form whatsoever without the written permission of the copyright owner.

2) The policy of Batteries Included being that of continuous product improvement, the contents of this manual are subject to change, from time to time, without notice.

3) All efforts have been made to ensure that the contents of this manual are correct; however, should any errors be detected, Batteries Included would greatly appreciate being informed.

4) Batteries Included can assume no responsibility for errors in this manual or their consequences.

CONTENTS

Introduction Page 1
How To Use This Manual Page 3

Part I: The ORACLE User's Guide Page 5

Chapter 1 - Getting Started Page 6

Section 1 - Hardware Page 7
Section 2 - The ORACLE System Page 8
Section 3 - Turning On Page 9
Section 4 - Taking Care of Your Disks Page 10
Section 5 - User Conventions Page 11
Section 6 - The ORACLE Keyboard Page 13
Section 7 - Backing-Up Your Program Page 14
Section 8 - Getting Loaded Page 16

Chapter 2 - The ORACLE Tutorial Page 18

Introduction Page 19
Lesson 1 - Formatting a Workdisk Page 21
Lesson 2 - The Empty File List Page 23
Lesson 3 - Creating a File Page 24
Lesson 4 - What Have You Done? - I Page 38
Lesson 5 - Setting a Filename Page 39
Lesson 6 - The Field List Page 41
Lesson 7 - Adding Data Page 42
Lesson 8 - What Have You Done? - II Page 53
Lesson 9 - Searching for Data Page 54
Lesson 10 - Modifying Existing Records Page 65
Lesson 11 - Setting Up the Printer Page 71
Lesson 12 - Designing Reports Page 74
Lesson 13 - Special Sorts Page 88
Lesson 14 - ORACLE Math Page 93
Lesson 15 - Printing Labels Page 104
Lesson 16 - Printing Forms Page 108
Lesson 17 - A Last Look at What You've Done Page 115

Chapter 3 - System Management Page 116

Section 1 - Exiting to BASIC Page 118
Section 2 - Disk Utilities Page 119
Section 3 - Sequential Files Page 122
Section 4 - Modifying Record Formats Page 125
Section 5 - Working With Multiple Drives Page 127
Section 6 - Full Disks/Splitting Files Page 129
Section 7 - Support Programs Page 130
Section 8 - Interfacing with PaperClip Page 132
Section 9 - ORACLE Security Page 133

Chapter 4 - ORACLE Design Page 136

Section 1 - Designing Your Own Applications Page 137
Section 2 - Creating Effective Formats Page 138
Section 3 - Direct Printer Commands Page 142
Section 4 - Multi-Page Printouts Page 143
Section 5 - Breakfields Page 145
Section 6 - Complex Searches Page 147
Section 7 - ORACLE Math Page 149

Part II - The ORACLE Reference Guide Page 152

Section 1 - Reference Guide to ORACLE Functions Page 153

Section 2 - Appendices Page 200

Appendix I - Inserting Your ROM Chip Page 201
Appendix II - ORACLE Special Keys Page 202
Appendix III - Sort Chart Page 203
Appendix IV - Security Access Chart Page 204
Appendix V - ORACLE Field Types Page 205
Appendix VI - Error Messages Page 207
Appendix VII - .HDR and .REL File Layout Page 209
Appendix VIII - Changing Default Values Page 211

Section 3 - Glossary Page 212

Section 4 - Index Page 218

Introduction to ORACLE

Come and consult the ORACLE -- Delphi's ORACLE, that is -- a powerful, easy-to-use database management system from the Delphi Systems Group.

What, you may ask, is a database management system? What, for that matter, is a **database***?

Well, the words may sound pretty fancy, but the ideas behind them are actually quite simple. A database management system is really nothing more than a kind of electronic filing system. In fact, it's very much like the familiar card file.

A card file is just a box with a series of cards inside. The cards hold information. At home, people often use card files to organize recipes, phone numbers, or cancelled checks. In offices, card files are often used to organize customer lists, billing information, or inventory records.

The crucial point about a card file is that the cards are not just chucked in any old way. Rather, they are kept in order -- so, the information they hold is kept in order, too. That's what makes the card file useful. It allows you to find the information you want, easily and quickly. It allows you to update the file -- to add new items to it and take old ones out -- without having to reorganize everything else.

Of course, the order of a card file is flexible. The order can be whatever is most convenient for the person using it. For example, a file of names and addresses would probably be ordered alphabetically. However, a company with a fleet of delivery trucks running on fixed routes might order the names of customers by route number instead. A file of recipes might be listed alphabetically, but the recipes might also be divided into categories -- 'Soups', 'Salads', 'Main Courses', 'Desserts', and so on -- and then ordered alphabetically within those categories. That way the file is useful, not only for finding a specific recipe, but also for browsing if you're not sure exactly what you want to make for a certain course.

To sum up: A card file is simply a method for storing information so that it is easy to find and work with.

Now, data is just a collection of facts -- raw, unordered information. A database is an electronic filing system for holding data. And a database management system is an electronic method for storing and ordering that data so it is easy to find and work with.

ORACLE is a database management system designed to work with your Commodore 64, CBM 8032, CBM 4032, or PET 2001 (upgraded to 32K) with a disk drive. Working together, your Commodore computer and ORACLE give you information-handling power and versatility that make card filing (not to mention a lot of other database management systems) look like something from the stone age.

- ORACLE allows you to create files, to add information to them, to update them whenever you want, to delete old information - all quickly and easily. Of course, that much a card file can do, too. But for ORACLE it's just the beginning.
- ORACLE allows you to redesign the layout of your files whenever you want, without having to re-enter all the old data.
- ORACLE gives you sophisticated, high-speed procedures for sorting your files into whatever order you want, and for searching your files quickly to find the information you need. ORACLE readily handles files thousands of items long.
- ORACLE lets you perform arithmetical calculations with your data. It allows you to look at and analyze data in revealing new ways.
- ORACLE lets you design reports for your database and print them out. It has built-in routines for designing and printing mailing labels and for filling out pre-printed forms.
- ORACLE allows you to use your data in word-processed documents, such as form letters and complex reports.
- ORACLE gives you an optional, three-level password security system, so you can keep your database entirely private or assign various levels of access to it.

In short, for just about any small business or home application, ORACLE brings sophisticated database management to the fingertips of the non-expert user. And because ORACLE is really no harder to handle than your average card file, you can learn to use it in an afternoon.

So what are you waiting for? Let's get going.

How To Use This Manual

You hear a lot about 'user-friendly' computers and software these days. But what good is a user-friendly system if the manual is a mad dog?

The ORACLE manual is different. Written in plain language, it has been specially designed for the non-expert user. It breaks down into two main parts.

Part I is The ORACLE User's Guide. It's the part of the manual that teaches you to use ORACLE.

> **Chapter 1,** Getting Started, deals with setting up equipment and loading the ORACLE program into your computer, and it offers a few helpful hints.
>
> **Chapter 2,** The ORACLE Tutorial, is the longest part of the manual. It deals with data management -- how to use ORACLE for storing and working with information. The Tutorial contains seventeen lessons, each covering a different ORACLE procedure. These lessons give you real hands-on experience in working with ORACLE. Using a sample database, the lessons take you step-by-step through all the everyday procedures you'll use in your own applications of ORACLE. By the time you've finished the Tutorial, you'll be an accomplished ORACLE user, with all the confidence and experience you need to strike out on your own.
>
> **Chapter 3,** System Management, deals with all the support procedures available to help you make the most of ORACLE. These include: disk utilities, ORACLE security, large-scale sorting, restructuring files, direct printer commands, interfacing ORACLE with word-processor programs, and the additional support programs included on your ORACLE disk.
>
> **Chapter 4,** ORACLE Design, deals with designing your own ORACLE applications and includes a detailed guide to the creation of screen layouts, printed reports, and search procedures.

Part II is The ORACLE Reference Guide. It's a comprehensive guide to the whole ORACLE system, designed to answer your questions and solve your problems quickly. It outlines all ORACLE procedures and contains glossaries of screen messages, BASIC commands used in ORACLE, special ORACLE key functions, and computer terminology.

The manual concludes with an index.

The best way to get started with ORACLE is to give the manual a quick onceover first. That will help to give you a feel for ORACLE even before you start working with it.

When you're ready to really get going, set aside some quiet time and work carefully through **Chapters 1** and **2.** Take them one step at a time, and take every step in turn because each lesson builds on the skills you'll have learned in the previous one.

Although you should be able to finish the Tutorial in an afternoon, it's not necessary to do everything at one sitting if you don't feel like it. You won't sacrifice anything by stretching the Tutorial out over two or three sessions if that's more convenient.

A Note on Disks

In the course of learning to use ORACLE you will be creating two new disks -- a backup disk for your ORACLE program, and a workdisk. Before starting **Chapter 1**, be sure you have two fresh disks ready.

A Note on Computer Terminology

It's impossible to write a computer manual without using some technical terms. In this manual, whenever such a term occurs for the first time, it will appear with an asterisk. Each of these asterisked terms is fully defined in the **Glossary of Computer Terms** near the end of this manual.

A Note on Printers

ORACLE can be used without a printer, but it only reaches its full potential on systems which include a printer. Many of the lessons assume that you have a printer connected. These are marked by **[P]** next to the title. If you don't have a printer, you needn't skip these lessons. You can still follow them on your screen.

Part I: The ORACLE User's Guide

Chapter 1 - Getting Started Page 6

Chapter 2 - The ORACLE Tutorial Page 18

Chapter 3 - System Management Page 116

Chapter 4 - ORACLE Design Page 136

Chapter 1 - Getting Started

Section 1 - Hardware Page 7

Section 2 - The ORACLE System Page 8

Section 3 - Turning On Page 9

Section 4 - Taking Care of Your Disks Page 10

Section 5 - User Conventions Page 11

Section 6 - The ORACLE Keyboard Page 13

Section 7 - Backing-Up Your Program Page 14

Section 8 - Getting Loaded Page 16

Section 1 - Hardware

ORACLE can be used with any Commodore computer with at least 32K of memory, and a 40- or 80-column screen. These include the Commodore 64, CBM 8032, CBM 4032, and PET 2001 (with 32K memory expansion).

ORACLE can be used with any Commodore single or dual disk drive. This includes models CBM 1541, 2031, 4040, 8050, 8250, 9060, or 9090.

ORACLE can be used with any printer that is properly interfaced to the above components.

Consult your hardware manuals regarding proper connection of your components.

Throughout the manual, we have assumed that 8032 systems have a dual disk drive (4040, 8050, or 8250) and that Commodore 64 systems have a single disk drive (1541).

Section 2 - The ORACLE System

For the Commodore 64

The ORACLE system for the Commodore 64 consists of the ORACLE manual, a **key***, and a **floppy disk*** containing a 40-column version of the master program.

Before you can use ORACLE on your Commodore 64, the key must be inserted in control port 1 on the right side of the computer.

For All Other Models

The ORACLE system for all other CBM computers consists of the ORACLE manual, a **ROM chip***, and a **floppy disk*** containing an 80-column and a 40-column version of the master program.

Before you can use ORACLE on your 8032 or 4032, the ROM chip must be inserted in the proper socket inside the computer. To insert the chip:

1) Turn off your computer and disconnect it from its power supply.

2) Remove the screws holding down the computer's housing, and prop the housing open. (See your computer manual for instructions.)

3) Consult **Appendix** I to find the right socket for the chip in your computer. The socket is #UD3 in model 2001, and #UD12 in other models.

4) Remove the ROM chip from its package. Making sure that the notch on the chip is facing the same way as shown in **Appendix I**, firmly seat the chip in its socket. Make sure all the pins are correctly inserted.

CAUTION

ROM chips are very delicate. Handle them with care. Take special care not to bend the pins.

5) Close the computer housing and refasten it.

And that's that. Before reconnecting your equipment to its power supply, make sure all components are turned off.

Section 3 - Turning On

Okay. Your ORACLE ROM chip or key is in your computer. Your computer, disk drive, and printer (if you've got one) are all hooked up correctly according to their respective manuals. Your machines are all plugged in.

It's time to get turned on. (If you're not exactly sure how, consult your component manuals.)

When the power goes on things start to happen.

As your disk drive powers-up it will whirr softly and its red and/or green lights will come on. That's normal -- don't worry about it.

A few seconds after you turn on your computer (and screen, if your screen is switched on separately, as with the Commodore 64), the screen will light up and display a few brief messages.

At the top is a message reminding you that your computer is a Commodore. Terrific. Very informative.

Next is a message telling you that the programming language used by your computer is **BASIC***, either BASIC 2.0 or BASIC 4.0, depending on which model you have.

Next is a line telling you how many **bytes*** of memory your computer has available.

Below that should be the word **READY**.

And below that a flashing **cursor***.

If your screen doesn't light up, or if it does but doesn't display the **READY** signal and flashing cursor, then you may have forgotten to turn something on, or you may not have things connected properly. Try turning everything off and then back on again. If that doesn't work, consult your component manuals.

Once you have a **READY** signal and a flashing cursor, your computer is ready for ORACLE.

However, we have a few more things to discuss first: disk handling, user conventions, the ORACLE keyboard, and backing-up your ORACLE master program disk.

Section 4 - Taking Care of Your Disks

For the protection of both your program and your data, disks must be handled with care. Follow these rules to keep your disks safe and sound:

- Keep each disk in its storage envelope when it is not in use.
- Store disks in a disk library case or some other rigid container.
- Never leave disks on top of computer or disk drive.
- Keep disks away from magnetic fields, such as those generated by transformers, electric motors, loudspeakers, and telephone bells.
- Never write on disk jacket or label with a lead pencil or ball-point pen. Fill out label before attaching it to jacket, using a felt-tip pen.
- Never expose disks to excessive heat or direct sunlight. Recommended temperature range for storage and use: 50 to 120 degrees F. (10 to 49 degrees C.)
- Never touch disk surface or centre hole: handle only by disk jacket.
- Never try to clean disk surface in any way.
- Never turn drive on or off with disks in the unit.
- Don't bend or handle disk roughly: insert it into drive unit gently.
- Don't remove disk from drive while program is running unless program tells you to do so.
- Never remove disk from drive while drive is whirring.
- Never attach notes or labels to disks with staples or paperclips.

Note: ORACLE will not work properly on disks with 'Write Protect Tabs'*.

Section 5 - User Conventions

In order to make this manual as clear as possible, we have established user conventions to tell you which keys to use on the computer's keyboard. There are four different user conventions in this manual.

PRESS: Whenever you see the term **PRESS:** in this manual, it means: Strike the single key specified immediately after the colon. For example:

PRESS:n

means strike the 'n' key on your keyboard. And

PRESS:RETURN

means strike the 'RETURN' key on your keyboard.

TYPE: means simply: Type the characters following the colon. For example:

TYPE:run

means type the word 'run' on your keyboard.

TYPE: The lazy dog lay down.

means type the full sentence 'The lazy dog lay down.' on your keyboard, including all capitals and punctuation.

ENTER: means type whatever is specified after the colon and then strike the 'RETURN' key. Striking 'RETURN' enters whatever was typed into the computer's memory. For example:

ENTER:run

means type the word 'run' and then strike the 'RETURN' key. And

ENTER: The lazy dog lay down.

means type the sentence 'The lazy dog lay down.' with all capitals and punctuation, and then strike RETURN.

<> Sometimes you will see pointed brackets enclosing a phrase after any one of the above three commands. Messages within the brackets are not meant to be typed, character for character. They simply describe what is to be typed. For example:

PRESS:<any key>

means strike any key on the keyboard. And

TYPE:<file name>

means type the name of the file you want to use.

ENTER:<file name>

means type the name of the file you want to use and then strike the RETURN key.

All keys pressed or characters typed in ORACLE in response to prompts* from the screen should be typed unshifted, regardless of how they appear on the screen. If a key should be typed shifted, we will always say so in the manual.

For example, the command to press shifted RUN/STOP key will read:

PRESS:(shift)RUN/STOP

This means: Hold down the shift key and press the RUN/STOP key.

Section 6 - The ORACLE Keyboard

In the ORACLE program the behavior of the cursor changes slightly depending on which mode of operation you're working in. We'll discuss these changes as we encounter them in the Tutorial. Otherwise, the cursor controls work exactly as they are described in your computer manual.

The CRSR keys move the CuRSoR right (unshifted) or left (shifted), down (unshifted) or up (shifted), without changing the text on the screen in any way.

The unshifted CLR/HOME key 'homes' the cursor (moves it to the top left-hand corner of the screen).

The shifted CLR/HOME 'homes' the cursor and CLeaRs the screen.

The space bar types a blank space.

The unshifted INST/DEL is the DELete key. It back-spaces and erases as it backs up.

The shifted INST/DEL is the INSerT key. It inserts blank spaces to the right of the cursor.

There is one rule which applies to all situations in ORACLE: **If you make a typing mistake, you can always change it by using the DEL key to DELete all characters back to the point where the mistake was made, and then type your entry again.**

Otherwise typing text onto the screen is pretty much like typing on a typewriter.

For more detailed information about the screen editing of text, consult your computer manual.

Section 7 - Backing-Up Your Program

Never use your ORACLE master program disk (the one suppied with your manual) as your everyday program disk. Instead, make a **backup*** disk (a copy) for daily use, and store the master in a safe place. That way, if anything happens to your everyday disk, you haven't lost your ORACLE program. You still have the master from which you can make another backup copy.

Making a backup program disk should be the first thing you do whenever you start to use any new program. Right now we're going to make a backup of your ORACLE master. To do it you'll need a fresh disk.

The procedure for backing-up a disk varies, depending on whether you have a single or a dual drive.

If you have a dual drive:

1) Make sure you have a **READY** signal and flashing cursor on the screen.

2) Insert your ORACLE master disk into drive 0 (zero). The large ORACLE label should be face up. (If you don't know how, consult your disk drive manual.)

3) Insert a fresh disk into drive 1 (one). Again, the disk label should be face up.

4) Now,

   ENTER:backup d0 to d1

   In other words, **TYPE:**backup d0 (zero) to d1 (one), then **PRESS:**RETURN

The cursor will disappear, your disk drive will start whirring, and the lights on the drive will go on. Your backup disk is complete when the **READY** signal and flashing cursor reappear. Remove your master from the drive, return it to its envelope, and store it in a safe place. Label your new disk "Oracle Backup" (making sure to write on a blank label before you stick it to the disk jacket).

If you have a single drive:

1) Make sure you have a **READY** signal and flashing cursor on your screen.

2) Insert a fresh disk into the drive. The disk label should be face up. (If you don't know how, consult your disk drive manual.)

   ENTER:open15,8,15

   In other words, **TYPE:**open15,8,15 and then **PRESS:**RETURN. Now,

   ENTER:print#15,"n0:orabackup,00"

   (That's zerozero). The disk drive will start to whirr.

3) While the disk whirrs for a while, get a fresh label and write "orabackup,00" on it with a felt-tip pen.

4) When the red light on your disk drive goes out after a minute or two, take the disk out of the drive, replace it in its envelope, and stick your new label onto the disk jacket. You've just **formatted*** your disk by giving it a name ("orabackup") and disk ID number (00) by which your computer will recognize it.

5) Now, insert your ORACLE master disk into the drive. The ORACLE label should be face up.

   ENTER:load"*",8

   The cursor will disappear from the screen and your disk drive will start to whirr.

6) When the flashing cursor reappears, remove the master disk from the drive, return it to its envelope, and store it away in a safe place.

7) Insert your newly formatted disk into the drive, and

   ENTER:save"Oracle 64.run",8

Make sure you use a shifted 'O'. When the **READY** signal and cursor reappear, your backup disk is complete. This single drive procedure copies your ORACLE program only. It does not copy any of the support programs on the master disk. We'll get to those later.

Section 8 - Getting Loaded

Finally, we're ready to load ORACLE into your computer. The things we've done up to now need to be done only once. From now on, whenever you want to work with ORACLE, all you have to do is load it into your computer from your backup disk.

For any model computer but the Commodore 64:

1) Make sure you have a ready signal and a flashing cursor on the screen.

2) Insert your backup ORACLE program disk into your disk drive. If you have a dual drive use drive 0 (zero).

   PRESS:(shift)RUN/STOP

4) A message that the program is loading appears. After a few moments a new prompt: **Do you want security codes? [n]** - appears. We'll look at ORACLE's security options in the **System Management** chapter of the manual. For now, however, we won't be using them. To disable the security system

   PRESS:RETURN

   A new prompt - **Save secured program** - appears. Below it is the **READY** signal and flashing cursor.

   ENTER:run

A new display suddenly appears. Across the top is ORACLE's copyright notice. Below that is ORACLE'S **Main Menu.** At the bottom is the message **Enter selection,** with the cursor flashing beside it. This is just ORACLE'S way of saying, "Hello."

If you don't get this display on your screen, make sure your ROM chip is inserted properly, re-load the program, and try again.

For the Commodore 64:

1) Make sure you have a **READY** signal and a flashing cursor on your screen.

2) Insert your backup ORACLE program disk in your disk drive.

   ENTER:load"Oracle 64.run",8

   Make sure you use a shifted 'O', and don't forget the blank space after the word 'Oracle'. On the screen the message **searching for Oracle 64.run** will appear, and shortly afterwards the message **loading.** There will be a pause while the disk drive whirrs and the program is loaded.

   If the message **file not found** appears, you have probably made a typing mistake. Try step 2 again.

3) When the flashing cursor reappears,

   ENTER:run

A new display suddenly appears. Across the top is ORACLE's copyright notice. Below that is ORACLE'S **Main Menu.** At the bottom is the message **Enter selection,** with the cursor flashing beside it. This is just ORACLE's way of saying, "Hello."

If for some reason, you do not get this display on your screen, make sure your key is inserted properly, and try again.

Chapter 2 - The ORACLE Tutorial

Introduction Page 19

Lesson 1 - Formatting a Workdisk Page 21

Lesson 2 - The Empty File List Page 23

Lesson 3 - Creating a File Page 24

Lesson 4 - What Have You Done? - I Page 38

Lesson 5 - Setting a Filename Page 39

Lesson 6 - The Field List Page 41

Lesson 7 - Adding Data Page 42

Lesson 8 - What Have You Done? - II Page 53

Lesson 9 - Searching for Data Page 54

Lesson 10 - Modifying Existing Records Page 65

Lesson 11 - Setting Up the Printer Page 71

Lesson 12 - Designing Reports Page 74

Lesson 13 - Special Sorts Page 88

Lesson 14 - ORACLE Math Page 93

Lesson 15 - Printing Labels Page 104

Lesson 16 - Printing Forms Page 108

Lesson 17 - A Last Look at What You've Done Page 115

Introduction to the ORACLE Tutorial

You're now ready to really start learning to use ORACLE. The Tutorial will show you how.

As we said in the introduction to this manual, the Tutorial is not just an explanation of how to use ORACLE; rather, its aim is to teach you ORACLE by actually getting you to use it on a sample **database***. Once you've completed the 17 Tutorial lessons, you'll be more than ready to strike out on your own.

The lessons are arranged progressively. Each of them builds on the skills you'll have learned in the previous one. Furthermore, each of the lessons is essential. So be sure to do all the lessons, and do them in order. You don't have to do them all in one sitting, but you should do them all eventually, and in the order they are presented.

At the end of some of the lessons you'll find a section entitled **Further Explorations**. This section contains some suggestions for experimenting with the function you've just learned to use. Although you won't be missing anything essential if you skip them, these further explorations can help you sharpen your ORACLE skills.

In the Tutorial you should always strike the keys exactly as the manual specifies, in accordance with the user conventions. However, everybody makes typing mistakes. When you do:

1) Even though the cursor controls may work a little differently in different modes, you can always use the DEL key to erase backwards and type over your mistake.

2) Any time you do something that causes a **reverse video*** **error message*** to appear without a flashing cursor, striking <any key> will take you back to the **Main Menu***.

3) Any time you see a flashing cursor on the screen, you can return to ORACLE's **Main Menu** if you

   PRESS:(shift)RUN/STOP

   on your 8032, or

   PRESS:f8

on your Commodore 64.

Throughout this manual both of the above commands will always be written

PRESS:<escape>

The <escape> function is an extremely useful routine in ORACLE. Remember it for later use. You can use it to return to the **Main Menu** from any screen in which you see a flashing cursor.

Note for 8032 users: The <escape> function has nothing to do with the ESC key on 8032 keyboards, which has no function in ORACLE.

Lesson 1 - Formatting a Workdisk

ON SCREEN: **Main Menu**

Okay. You've got ORACLE loaded (If not, see **Getting Loaded** for instructions) and the **Main Menu** is on your screen. Before we start exploring ORACLE, we have to prepare a **workdisk*** for the sample database we'll be working with.

What, you're asking, is a workdisk?

Well, the disk you have in your drive right now is your backup program disk. It contains the ORACLE program. There's still plenty of room on it, and, if you wanted to, you could also use it to contain the sample database.

However, it's best not to use your program disk for storing anything except the ORACLE program. And you should never use your master ORACLE disk for storing any information other than its original programs. Storing information on the master disk may destroy it entirely.

The rule, then, is that when you're working with a database, you'll need to create a workdisk or datadisk, one that's just used for holding your ORACLE data files.

Preparing a disk for use is called **formatting*** a disk. Formatting means giving the disk a name and an ID number by which the computer can recognize it. Those of you working with single drives have formatted a disk once already when you were creating your ORACLE backup. At the time, though, you weren't yet inside ORACLE and had to use **BASIC***. Now we're going to format a disk using ORACLE. This is the procedure you should use whenever you make an ORACLE workdisk.

To format your workdisk:

1) Make sure ORACLE is loaded in your computer and that the **Main Menu** is on the screen.

2) Remove your backup program disk from the drive, and put it away. You won't be needing it until you have to load ORACLE again.

3) Put a fresh disk into your drive. If you're using a dual drive, use drive 0 (zero).

4) Look at the **Main Menu.** It contains six options. The routine for formatting a disk is contained in **5 - Disk Utilities.**

PRESS:5

5) On screen the **Disk Utilities** menu appears. It has seven options. Formatting is **5.** Again,

PRESS:5

6) A **prompt*** appears on screen. A prompt is just a message from the computer, telling you what to do next. The prompt says: **Enter disk name, two char. ID.** It is telling you to type a name for your disk onto the screen, followed by a comma, followed by any two characters to serve as the disk's ID number. Later, in the **System Management** chapter, we'll look at the rules governing disk names and ID numbers. For the moment we'll just do the following:

ENTER:sample,sa

Remember, regardless how these characters appear on the screen, you are typing unshifted keys. And don't forget to strike RETURN.

7) After you hit RETURN, a **reverse video*** message will appear at the bottom of the screen asking you to **please standby.** Your disk drive will start up and a light will go on. When the light goes out, the screen will return to the **Disk Utilities** menu. The formatting is finished.

8) To return to the **Main Menu**

PRESS:7

Lesson 2 - The Empty File List

ON SCREEN: Main Menu

You're back in the **Main Menu,** and you're the proud owner of a newly formatted workdisk named 'sample'. If you haven't done so already, prepare a label for your workdisk, identifying both its name and ID number. It's also a good idea to indicate on the label exactly what is contained on the disk. For this disk you might write 'ORACLE Tutorial Data'.

So far, there's nothing on your disk -- at least no files, just the disk name. Want proof?

1) Go back to the **Disk Utilities** menu.

2) Ask ORACLE to show you all the files on your disk. You do this by choosing disk utility **1 - Show All Files.**

   PRESS:1

3) The utility menu disappears. In its place is a screen displaying your disk's name on the top line in reverse video, the number of **blocks*** still available on the disk, and a prompt: **Strike any key to continue.** There are no filenames listed because you haven't created any files yet. You soon will, though. A few lessons down the road we'll come back to this file list to see how it has changed.

4) Return to the **Disk Utilities** menu.

   PRESS:<any key>

5) Return to the **Main Menu.**

Lesson 3 - Creating a File

ON SCREEN: Main Menu

Now, it's time to give your imagination a little exercise.

Let's pretend that you belong to a community organization that's trying to raise money for a new playground. As one of the organization's champion fundraisers, you've asked local teenagers to volunteer to sell boxes of candy and cookies. The neighborhood has been divided up into a number of sales routes. Your job is to supervise the volunteers and oversee the program.

To keep track of everything you'll need a filing system.

You could do this with a card file if you wanted to. You would assign a section of the file to each volunteer. Each section would consist of one or more cards, depending on how much space you needed. On each card there would be a space for each item of relevant information -- name, address, age, route number, number of boxes sold, and so on. Then you would arrange the records in a file box -- perhaps alphabetically by name, or numerically by route number. You'd pick whatever order was most useful to you.

As it happens, though, you own a Commodore computer. And a copy of ORACLE. And since ORACLE can give you much more flexibility than a card file, you'll use it to create your filing system.

As we said, there are ways in which ORACLE is a lot like a card file.

As you would with a card file, you assign a section of your database file to each volunteer. In ORACLE, this section is called a **record***. Just as each volunteer's record in a card file may consist of more than one card, each volunteer's record in ORACLE may consist of more than one full screen of information. Each screen of information is called a **page***.

On each card of a card file, you assign a space for a specific item of information. In ORACLE, you also assign a space in each of your records for a specific item of information. This space is called a **field***.

And finally, in a card system, you'd keep all the records containing related information in the same place, probably in a small box. You'd call this collection of records a file. In ORACLE, you also keep all the records containing related information in the same place -- on a disk. This disk collection of related records is also called a **file***.

If you were constructing a card file, your first step would be to design the layout of each card -- just where on the card each bit of information should go. Laying out each card in exactly the same way makes it easy to find the information you're looking for, to compare cards, or to sort them when necessary.

Your first step in creating an ORACLE file is also to design the layout of each record -- just where in the record each field should go. That makes it possible for ORACLE to look for, compare, and sort the information in your file. In ORACLE, this record layout is called the **record format***.

In this lesson you will create a record format for your volunteer file. The aim of this lesson is to show you how to create record formats, not to teach you how to design them. Therefore, at this stage, we won't be explaining why we're doing what we're doing, just how to do it. Later, some of the reasons for the design of the format will start to become apparent. For complete information on the design of record formats see the **ORACLE Design** chapter of this manual.

Our record format will contain 14 fields. Each of the fields will hold a unique piece of information about one of our volunteers:

1. volunteer's last name
2. volunteer's first name
3. volunteer's social security number
4. volunteer's age
5. volunteer's street address
6. volunteer's apartment number
7. volunteer's city
8. volunteer's zip code
9. date the volunteer started working
10. volunteer's route number
11. number of boxes of candy the volunteer has sold
12. number of boxes of cookies the volunteer has sold
13. date we last updated the volunteer's record
14. comments about the volunteer's work

Let's get started.

1) Get the **Main Menu** on the screen.

2) The **Main Menu** option for creating a new data file is **2.**

   PRESS:2

3) A new screen appears. The top of the screen displays **Create Page 1.** The bottom of the screen displays the prompt: **Enter 1 for old or 2 for new format.** Since we want to create a new format,

   PRESS:2

4) A new prompt appears: **Enter <up arrow> to start field and <left arrow> when done.** The cursor is flashing in the upper left-hand corner of the screen. You can now move the cursor anywhere on the screen, using the cursor control keys. (Try it.) The prompt is telling you that if you strike the <up arrow> key you will be starting a field (a space to hold information) wherever the cursor is flashing. It's also telling you to strike the <left arrow> key when you've finished creating all the fields you want on the page.

   But fields are not the only things you'll want in your record layout. You'll also want to name each field, in a way that will tell you (or any other user) exactly what information the field is supposed to hold. To name your fields, you have to use **out-of--field characters***, i.e., characters which appear on the screen as a part of a record format but which are not inside a field.

   Our first field is going to hold our volunteer's last name, so we'll call it (appropriately enough) 'last name'. Make sure your cursor is in the upper left-hand corner of the screen. You can get it there quickly with

   PRESS:CLR/HOME

5) Now

   TYPE:last name....

   Don't forget to put in all the periods. And remember you can always correct typing mistakes with the DELete key.

6) With 'last name....' typed in, you're ready to create the 'last name' field.

   PRESS:<up arrow>

7) A vertical bar appears where the cursor was, and the cursor moves over one space. The bar is called the **open field marker***. The new prompt line displays: **Enter Length:** followed by the number of spaces still available on this screen for the field. (As we progress, you'll notice this number decreases.) We want our 'last name' field to be 20 characters long. So

   ENTER:20

8) Twenty spaces to the left of the open field marker another vertical bar appears. This is the **close field marker***. The cursor is blinking to its right. (If all you've got is the number '20' next to the open field marker followed by the cursor, you've forgotten to strike RETURN. Remember what **ENTER:** means.)

   The prompt reads: **Alpha, Numeric or Key field?.** These are different field **attributes***. When it comes time to enter your data, an **alpha field*** will accept only alphabetic characters, asterisks, and question marks; a **numeric field*** will accept only numbers, asterisks, question marks, hyphens, periods, and backslashes. A **key field*** is one that automatically gets sorted -- either alphabetically or numerically, depending on the kind of data it contains.

   We want our 'last name' field to be an alpha/key field. The prompt means type 'a' for alpha, 'n' for numeric, and 'k' for key. So, to specify an alpha/key field

   ENTER:ak

   Remember: All letters typed in response to a screen prompt should be typed unshifted.

   If you were paying attention to the screen as you pressed RETURN, you noticed some changes. The 'a' and the 'k' you typed in have disappeared. At the same time the open field marker became thinner and moved a little to the left. This indicates that our field is an alpha field. As well, the close field marker turned into a checkered bar, which means that the field has been closed.

9) Now, we want to create our next field, the 'first name' field. Generally speaking, it is possible to put more than one field on a line if you've got the space. However, this time we'll start the next field on the next line down. To get the cursor onto the next line

   PRESS:RETURN

10) Your cursor should now be on the second line at the left of your screen. Our 'first name' field is going to be almost the same as our 'last name' field, so setting it up will be much the same.

   TYPE:first name...
   PRESS:<up arrow>
   ENTER:20

   Don't forget to hit RETURN. Your cursor should be flashing to the left of a vertical bar directly under the close field marker of the first field. The prompt is asking if you want an alpha, numeric, or key field. Since we want an unattributed field here, and don't want to specify any of the choices,

   PRESS:RETURN

   Once again notice that the field markers change, but because this is an unattributed field, the open field marker is a little different from the one in the 'last name' field. Our second field is finished. To get to the next line

   PRESS:RETURN

11) Our third field is for social security number. Social security numbers take the form XX-XXX-XXXX. This means that we have to make the field long enough to include the dashes as well as the digits. So we want an unattributed field that is 11 spaces long. To create the field

   TYPE:SS#.........
   PRESS:<up arrow>
   ENTER:11
   PRESS:RETURN

   The field is finished. To go to the next line

PRESS:RETURN

12) The fourth field is for age. It will be an ordinary field, 2 spaces long.

TYPE:age..........
PRESS:<up arrow>
ENTER:2
PRESS:RETURN

And to go to the next line

PRESS:RETURN

13) Our first four fields are finished. Now, to help make the screen more readable, we'll put a horizontal line to separate this first block of fields from the following block. To do this we will just underline continuously across the screen from the left edge to the right edge. On the Commodore 64 the underline key is the <British Pound> key. On other machines it is the <backslash> key. To underline

PRESS:<British pound>

or

PRESS:<backslash>

over and over again, as many times as it takes you to go right across your screen. When you have finished

PRESS:RETURN

Let's pause for a moment and take a look at what you've done so far. Your screen should look something like Fig. 2.1.

```
last name....[                    ]
first name...[                    ]
SS#..........[           ]
age..........[  ]
____________________________________
```

Fig. 2.1

If your screen doesn't look at all like Fig. 2.1, return to the Main Menu with

PRESS:<escape>

(Remember: <escape> is f8 on the Commodore 64 and (shift)RUN/STOP on other machines.) Then go back to step #1 on page 26 and try again.

If everything's okay, though, let's move along.

14) Our next block will consist of four fields -- 'street' (20 spaces long), 'apartment #' (4 spaces), 'city' (15 spaces), and 'zip code' (5 spaces). They will all be unattributed fields. You should be able to do these a little faster than you did the last four. To create the 'street' field

    TYPE:street.......
    PRESS:<up arrow>
    ENTER: 20
    PRESS:RETURN
    PRESS:RETURN

    To create the 'apartment #' field move down a line and

    TYPE:apt..........
    PRESS:<up arrow>
    ENTER: 4
    PRESS:RETURN
    PRESS:RETURN

    To create the 'city' field move down a line and

    TYPE:city.........
    PRESS:<up arrow>
    ENTER: 15
    PRESS:RETURN
    PRESS:RETURN

    And to create the 'zip code' field move down a line and

    TYPE:code.........
    PRESS:<up arrow>
    ENTER: 5
    PRESS:RETURN
    PRESS:RETURN

    To finish, underline the whole block of fields, just as you did with the first block, and then

PRESS:RETURN

Once more, let's take a look at what you've done. Your screen should look like Fig. 2.2.

```
last name....[                    ]
first name...[                    ]
SS#..........[           ]
age..........[  ]
_____________________________________

street.......[                    ]
apt..........[    ]
city.........[               ]
code.........[     ]
_____________________________________
```

Fig. 2.2

Notice how neatly the fields are lined up. You can see why we've been using all those periods.

If your screen does not resemble Fig. 2.2, you are going to have to make some corrections.

This time, you don't want to go back to the **Main Menu** and start over because that would mean losing your already corrected first block. We'll have to look at some new ways of making corrections.

One way is with the cursor control keys. As long as you don't have any unfinished fields on your screen, you can use the cursor keys to move the cursor to any spot on the screen. To correct a spelling mistake, simply type over it. Use the INST/DEL key to insert and delete spaces where needed -- even inside a field to change the field length. Try it.

To change the attribution of the field (alpha, numeric, key, or unattributed), however, you must create the field over again. Move your cursor to the start field marker, begin the field-creation process again with **PRESS:**<up arrow>, and make your attribution change at the normal spot, after closing the field. You'll see that the **Enter Length:** prompt indicates one less than your current field length. Nevertheless, you can still select the full length of your original field.

15) Once you've made any changes you need, you're ready to start your last block of fields on this page. The block will contain five fields:

- o An unattributed 'date started' field, 6 spaces long.
- o A numeric/key 'route #' field, 2 spaces long.
- o An unattributed 'candy' field, 4 spaces long.
- o An unattributed 'cookies' field, 4 spaces long.
- o An unattributed 'last update' field, 6 spaces long.

Those are the field specifications. We're going to let you try these fields on your own. Start your fields down one row from your last underline, and don't forget to add enough periods after each field name to make the open field markers line up properly.

Good luck!

All done already? Okay then. If you've got everything right, your screen should now look like the one in Fig. 2.3

```
last name....[                    ]
first name...[                    ]
SS#..........[           ]
age..........[  ]
____________________________________

street.......[                    ]
apt..........[    ]
city.........[               ]
code.........[     ]
____________________________________

date started.[      ]
route #......[  ]
candy........[    ]
cookies......[    ]
last update..[      ]
```

Fig. 2.3

If your screen doesn't resemble Fig. 2.3, make any corrections that you need to.

This page of our record format is almost done. But to make it a little easier to read, we should double-space the third block. To do this we'll use the ORACLE screen editing function that allows you to insert and delete whole lines.

16) Move your cursor to the letter 'r' in the word 'route', and

    PRESS:RUN/STOP

    A new screen prompt appears: **(+)=INST, (-)=DEL LINE.** If you now strike the <plus> key, ORACLE will insert a blank line at the cursor position, pushing all the remaining lines down one. If you strike the <minus> key (Don't do it!), ORACLE will erase the line the cursor is on and move all the lines below it up one.

    You want to insert a line. So

    PRESS:<plus>

    The 'route #' line and all the lines below it move down one. To see how the delete line function works

    PRESS:<minus>

    Now you're back where you started.

17) Use the insert-line function to double-space all the lines in the third block. Between insertions you'll have to get out of the function in order to move your cursor to the next position. To get out of the insert/delete-line function

    PRESS:<any key>

    except, of course, the <plus> or <minus> key.

18) Now there's just one more thing to do and our page will be complete. Move your cursor to the right side of the close field marker of the 'candy' field. Space over two spaces, and

    TYPE:BOXES SOLD

Now move your cursor to the right side of the close field marker of the 'cookies' field, space over two spaces, and

TYPE:BOXES SOLD

If everything has gone well, your screen is finished. It should look like Fig. 2.4

```
last name....[                    ]
first name...[                    ]
SS#..........[           ]
age..........[  ]
------------------------------------
street.......[                    ]
apt..........[    ]
city.........[               ]
code.........[     ]
------------------------------------
date started.[      ]

route #......[  ]

candy........[    ]  BOXES SOLD

cookies......[    ]  BOXES SOLD

last update..[      ]
```

Fig. 2.4

If it's still not right, make whatever corrections needed.

Once everything is okay, you're ready to save the format (you're finally getting to use that <left arrow> key).

19) To save your format

PRESS:<left arrow>

If you've made a mistake and your screen does not contain an equal number of open and close field markers, the message: Error - 'open' & 'close' unmatched will appear on the prompt line. The cursor will be flashing at the point on the screen where the problem lies, or on the bottom line if you have left off the last close field marker. Correct the mistake by either DELeting the extra close marker or by recreating the problem field(s). Once your screen looks okay, use <left arrow> again to finish.

If ORACLE agrees that your screen is okay, a new prompt appears: **Record length=0119 all fields o.k.? [y]**. The cursor is flashing over the letter 'y' (indicated in this manual by the square brackets). The prompt is telling you that your fields take up 119 spaces, and asking you if everything on the screen is okay. Take a look. If you find a mistake, you can

PRESS:n

for 'no'. The cursor jumps to the upper left of the screen, the prompt changes back to the original 'start field' prompt, and you can change whatever you like. Strike <left arrow> again when you're done.

However, if everything is okay, the cursor flashing over the letter 'y' means that if you

PRESS:RETURN

you will automatically be answering 'yes'.

ORACLE often displays a flashing cursor over one of the possible responses to a prompt (indicated in this manual by square brackets). By pressing RETURN at this point, you automatically accept that response. This is called the default* response. You are never required to accept the default. ORACLE is just trying to make it easier for you to make the most likely responses to its prompts.

A new prompt appears: **Would you like another page? [n]** You would.

PRESS:y

Page 2 of your record now appears, nice and clean and blank. With ORACLE you can create records up to nine pages long, with up to 99 fields altogether. However, for our volunteer files, we need only one more page, and it is only going to have one field on it.

20) You want to create a field 75 spaces long for remarks, so

TYPE:remarks.....
PRESS:<up arrow>
ENTER: 75
PRESS:RETURN

Your field is complete. Notice how it is possible for a field to occupy more than one line.

21) You're finished with this page, so

PRESS:<left arrow>

to accept the format. In response to the new prompt

PRESS:RETURN

to default to 'yes', and then

PRESS:RETURN

to default to 'no' because you don't want any more pages.

An entirely new screen appears, entitled, importantly, **Statistics of Creation.** The first line tells you the total number of characters in your record (the cumulative length of all your fields), in this case **194.** If you don't have 194, you've made a mistake in one of your fields.

The next couple of lines tell you the number of pages in your record format **(2)** and the total number of fields **(14)**.

The next line tells you approximately how many records of this size you can get onto your disk (in this case **512.**)

And the final line is a prompt asking you to enter a filename.

22) You have to give your file a name so that both you and ORACLE will be able to identify it. In ORACLE, a filename can be any **string*** of letters or numbers up to ten characters long. It can be punctuated only by a hyphen. ORACLE will not accept any other characters in a filename. Our filename will be 'treats'.

ENTER:treats

Notice that ORACLE will not write anything except capitals in a filename, even though you are typing unshifted keys.

After you strike RETURN, a new message appears, advising you that ORACLE is creating your file. When the file is created, you will automatically be returned to the **Main Menu.**

And that's how you create a record format for an ORACLE file. It's all downhill from here.

Lesson 4 - What Have You Done? - I

ON SCREEN: Main Menu

You've just finished creating your first ORACLE file. The question is: Where is it? Is your file really saved somewhere on the 'sample' disk? Well, see for yourself.

Go to the **Disk Utilities Menu** and select the **Show all files** option. When you did this before in **Lesson 2,** all that was on the screen was the disk name, the number of **blocks*** remaining on the disk, and a **Strike any key...** prompt.

The screen should look a little different now.

The number of blocks remaining in the **BLOCKS FREE** message is fewer than before, and there are two new items on the screen. These items are filenames.

One of these is **TREATS.REL.** The **.REL** stands for 'relative'. **TREATS.REL** is your actual data file which will store the volunteer information you enter.

The other is **TREATS.HDR.** The **.HDR** stands for 'header'. **TREATS.HDR** stores the actual record format of your TREATS file.

Whenever ORACLE presents information to you, it recombines the header and the relative files, so you can see your records just as you originally laid them out.

Now that you've seen what you have done, you can return to the **Main Menu.**

Lesson 5 - Setting a Filename

ON SCREEN: Main Menu

In order to be able to enter or modify data, or even just to view your records, it's necessary to tell ORACLE which file you want to work with. This may seem a little silly right now because there is only one file, **TREATS**, on our workdisk. However, if there were several files on your disk, ORACLE (smart as it is) wouldn't be able to read your mind to see which one you wanted to use. So even if your disk has only one file on it, you still have to be specific.

Selecting a datafile is called setting a filename.

1) The Main Menu option for setting a filename is **1 = Set data-file name.**

   PRESS:1

2) The **Set Filename** screen appears. On it is a list of the relative files on the disk, and the prompt **Enter Filename ?** with a flashing cursor beside it.

   The only file on our list is **TREATS.REL.** In the **Set Filename** mode, only relative files are presented.

   To set a filename, you enter only that part of it that appears before the **.REL.** To save you keystrokes, ORACLE automatically assumes the **.REL,** but let's see what happens if you try to enter the full name. Remember to type in the filename unshifted.

   ENTER:treats.rel

   After a moment, ORACLE presents an error message in reverse video: **62, file not found,00,00.** That's because ORACLE thinks you're looking for **TREATS.REL.REL,** and no such file exists. Whenever you ask for a file that doesn't exist on your disk, you'll get this error message. As with any reverse video message, you proceed with

   PRESS:<any key>

   You're now back at the original **Set Filename** screen.

3) Now let's set our filename properly.

ENTER:treats

If you make any mistakes typing in the name, use the DELete key to erase and start over.

After the filename is set, there is a short pause and ORACLE returns you to the **Main Menu.** The filename will now remain **TREATS** until you change it.

Note: If you try to enter any data-management function without a filename set, ORACLE automatically takes you to the Set Filename screen. If you exit this screen without setting a filename (by using PRESS:<escape>), you unset the current filename. Once you are in the Set Filename screen, you must always enter the filename you want to work with.

Lesson 6 - The Field List

ON SCREEN: Main Menu

Now that you've got a filename set, you can look at a list of the fields contained in your record format. You get to the field list option by selecting **4 = Searching and Reporting** from the Main Menu.

PRESS:4

List fields is option **6.**

PRESS:6

A new screen appears, containing a list of all the fields you have defined. Down the left side of this screen is a column of numbers (called **field numbers***). These field numbers are assigned by ORACLE according to the positions of the fields in the record layout you originally created (reading left to right, top to bottom).

NOTE: If you get a series of 'W's as a field name, you've forgotten to leave the blank lines in your format.

The next column lists all your field names, and in the final column shows the length of each field.

This **List fields** display is very useful for searching and reporting procedures, when you'll need to know the field numbers and lengths in order to design printouts.

PRESS:<any key>

to return to the **Search and Report** menu. Then return to the **Main Menu.**

Lesson 7 - Adding Data

ON SCREEN: Main Menu

With your record format created and your filename set, you're ready to fill your file with data.

Before we begin, there's one general precaution we should state. There's an acronym that gets used a lot in computer circles: GIGO. It stands for 'Garbage In, Garbage Out'. What it means is that your program is only as good as the data you put into it. If the input is incorrect, then when you ask the computer to work with it -- to do calculations or print reports -- the output will be incorrect too. Garbage in, garbage out.

GIGO is as true in ORACLE as it is anywhere else. The information in your ORACLE reports will print out exactly as you typed it into the field. If you want it to be right, you have to type it in right. The same is true for calculations. If you want your results to be right, make sure the numbers you enter are right.

In this lesson you're going to learn how to put information into ORACLE files. You'll be filling ten records with data. Enter your data exactly as specified. Remember GIGO.

1) The **Add new record** mode is listed under menu option 3 **= Modify an existing file.** So

   PRESS:3

2) A new screen appears displaying your record format as you laid it out in **Lesson 3**. The screen is titled **File Maintenance.** At the bottom is the prompt **Enter Exit, Add or Update** - with a cursor flashing beside it. You want to add records, so

   PRESS:a

3) A new prompt appears: **File or Keyboard? [k]** with the cursor flashing over the letter 'k' (the default response) beside it. By pressing 'f', you would select 'file' -- the ORACLE option for adding data to your records automatically from outside files. We'll explore this option later. For now, and whenever you want to type data into your records from the keyboard, default to 'k' with

PRESS:RETURN

4) The screen is now titled **ADD RECORD**, and the prompt is **Enter record data (<left arrow> key to finish).** The cursor is blinking in the first space in your first field.

CAUTION: Press the <left arrow> key only when you are finished entering all the data to the page. Pressing <left arrow> instructs ORACLE to accept the current page as complete and to move to the next page of your record. When you press <left arrow> from the last page of your record, ORACLE will automatically write the record to the disk. To avoid writing incomplete records, be sure you don't press <left arrow> until you're really ready to.

5) Before you begin to add data, familiarize yourself with the action of the editing keys in this mode. Experiment with the INST/DEL key and the CRSR keys.

   Note that INST/DEL will not change the size of the field or erase field markers, and that CRSR left-right will not carry you past a field marker in either direction.

   To get from one field to another, you use the RETURN key or the CRSR up-down key. Try it. Note that both RETURN and CRSR down will loop the cursor back to the first field from the last field.

   Finally, if you strike CLR/HOME from anywhere on the page, the cursor moves to the first spot in the first field.

6) Okay. Now you're ready to start adding data. Here's the information on our first volunteer:

Name: Maria California
Address: 64 Commodore St., Apt. 12
Compucity, USA
12345
SS#: 28-456-8878
Age: 14
Started: Feb. 12, 1983
Route: 7
Candy boxes sold: 14
Cookie boxes sold: 23
Last update: Mar. 15, 1983
Remarks: none

Make sure your cursor is blinking in the leftmost space of the 'last name' field. Remember, we made this field an alpha field. Just for an experiment, try to type in some numbers. Notice what happens -- absolutely nothing.

Now let's enter our data. The first field is the 'last name' field, so

TYPE: California

It's important that you type all words into your fields **left-justified*** in the field; that is, as far left in the field as possible. Remember also to capitalize where you have to. GIGO!

If you've got the last name correctly typed in,

PRESS:RETURN

to get your cursor in the next field, and

TYPE: Maria

to enter the first name. Now

PRESS:RETURN

to get into the 'SS#' field. Remember, we wanted to include the hyphens as well as the numerals.

TYPE:28-456-8878

and

PRESS:RETURN

to jump down to the fourth field. To enter the age

TYPE:14

and

PRESS:RETURN

to jump down to the 'street' field. To enter the street address

TYPE:64 Commodore St.

Be sure to type all spaces and punctuation exactly as you want them to appear.

We've now reached the 'apartment #' field. We've made 4 spaces available, figuring that 4 characters is the likely maximum length of any apartment number. In this case though, our number is shorter than the space available. When that was the case with words, the correct way to put them into the field was left-justified. With numbers, the exact opposite is correct.

All numbers must be right-justified* in their fields for ORACLE to sort them properly.

All words must be left-justified in their fields for ORACLE to sort them properly.

You can right-justify numbers in one of two ways. You can leave spaces before the number, or you can type 0's (zeros) before the number. For example, apartment #12 could be entered as **0012** or as **<space><space>12.** But you must not mix spaces and 0's (zeros) in the same field of a record (that is, you must not enter one person's apartment number as 0012 and another's as <space>301). If you do, ORACLE will get confused when trying to sort the field. You can, however, use 0's (zeros) in one field and spaces in another field in the same record.

When printed, the field will look exactly as you typed it -- with either zeros or spaces.

For our 'apartment #' field we'll justify using spaces. To enter Maria California's number

TYPE:<space,space>12

and then

PRESS:RETURN

to jump down to the 'city' field. To fill this field

TYPE: Compucity, USA

and

PRESS:RETURN

to jump down to the 'zip code' field. This field contains a number, but since zips have five digits and the field has five spaces, we don't need to worry about justification.

TYPE:12345

and

PRESS:RETURN

to finish off our second block of fields.

We come now to the 'date started' field. You may have thought it strange that we created the 'date' fields with only six spaces. However, one very convenient way to enter dates in an ORACLE field is to change them into six-digit numbers. The first two digits are for year -- 1983=83, 1941=41, and so on. The second two digits are for month -- January=01, February=02, December=12, etc. And the last two digits are for the day - the first=01, the seventh=07, the twentieth=20, and so on.

The YYMMDD format allows ORACLE to sort dates very easily. To enter Feb. 12, 1983, in the 'date' field

TYPE:830212

and then

PRESS:RETURN

to move on.

The 'route #' field is a numeric field. It won't accept letters, just as our 'last name' field wouldn't accept numbers. To enter the route number

TYPE:<space>7

PRESS:RETURN

to get to the 'candy' field. Here, we're going to use 0's (zeros) instead of spaces to justify the 'candy' and the 'cookie' fields. Later, you'll see how this will affect your printouts. From here on in everything is straightforward. To enter all the remaining data for this page,

TYPE:0014
PRESS:RETURN
TYPE:0023
PRESS:RETURN
TYPE:830315

7) Your cursor is now in the last field on the page. If you strike RETURN now, it will simply loop up to the first field again. If you're satisfied that everything is entered correctly, you can leave this page of the record with

PRESS:<left arrow>

as the prompt indicates.

8) The second page of your record now appears, with the **Enter record data** prompt at the bottom of the screen. You could now enter any remarks. However, there are no remarks for Maria, so we will just leave the field empty. Since that means all the data has been entered,

PRESS:<left arrow>

The prompt line reads: **Writing Record.** ORACLE is now **writing** the data you've just entered into the **TREATS.REL** file.

9) ORACLE takes only a few seconds to write the record. When it's done, the prompt line reads: **Add another record (y/n) ? [y]** Since we do want to add another record

PRESS:RETURN

to default to 'yes'. The **ADD RECORD** screen reappears. The format is empty. The cursor is flashing in the first space of the first field, and you can now add a second record.

10) Here is our information on our second volunteer, Jose California, Maria's twin brother.

```
              Name: Jose California
           Address: 64 Commodore St., Apt. 12
                    Compucity, USA
                    12345
               SS#: 29-871-6524
               Age: 14
           Started: Feb. 12, 1983
             Route: 9
 Candy boxes sold: 11
Cookie boxes sold: 1
       Last update: Mar. 28, 1983
           Remarks: Suspected of eating cookies
```

You can type in this record exactly as you did the last, with one small difference. Wherever a field repeats exactly the data for the same field in the previously added record, you won't have to type it again. Simply strike the <up arrow> while you're in the field, and ORACLE repeats the data for you.

Here's how it works. In the first field of this record, Jose and Maria have the same last name, so all you have to do to is

PRESS:<up arrow>

'California' will appear on the screen, and the cursor automatically jumps to the next field.

12) Enter the remaining data into this record with the following (remember, **ENTER:** means **TYPE:** and then **PRESS:**RETURN).

ENTER: Jose
ENTER:29-871-6524
PRESS:<up arrow>
PRESS:<up arrow>
PRESS:<up arrow>
PRESS:<up arrow>
PRESS:<up arrow>
PRESS:<up arrow>
ENTER:<space>9
ENTER:0011
ENTER:0001
TYPE:830328

Check your page for mistakes. If you find any, take the cursor to the problem field, and make your corrections. When you are satisfied,

PRESS:<left arrow>
TYPE: Suspected of eating cookies.

Again check for mistakes. When you are satisfied,

PRESS:<left arrow>

12) When ORACLE asks you if you want to add another record to the file,

PRESS:RETURN

to default to 'yes', and you're ready to go on with the next record.

13) We'll leave you now to fill in the last eight records on your own. Fill them out in exactly the order they are presented. For all volunteers, the 'city' field and the 'date started' field are the same. As long as you don't leave the **Add Record** mode, you'll be able to fill these fields in every record simply by using the <up arrow> key.

Don't forget to justify all your entries correctly.

Here's the information:

Name: Allan Fliegel
Address: 4040 Dual Dr.
Compucity, USA
12389
SS#: 65-254-9876
Age: 18
Started: Feb. 12, 1983
Route: 1
Candy boxes sold: 122
Cookie boxes sold: 67
Last update: Apr. 1, 1983
Remarks: none

```
               Name: Annie Ablative
            Address: 1541 Diskdrive Lane
                     Compucity, USA
                     12367
                SS#: 38-198-7365
                Age: 17
            Started: Feb. 12, 1983
              Route: 8
   Candy boxes sold: 56
  Cookie boxes sold: 34
        Last update: Mar. 11, 1983
            Remarks: none


               Name: Alicia DeSoto
            Address: 12 Floppy Diskway
                     Compucity, USA
                     12368
                SS#: 65-981-6729
                Age: 16
            Started: Feb. 12, 1983
              Route: 6
   Candy boxes sold: 46
  Cookie boxes sold: 67
        Last update: Feb. 28, 1983
            Remarks: none


               Name: Albert Galapagos
            Address: 64 Keyboard Pl.
                     Compucity, USA
                     12356
                SS#: 72-781-5618
                Age: 19
            Started: Feb. 12, 1983
              Route: 10
   Candy boxes sold: 107
  Cookie boxes sold: 29
        Last update: Mar. 11, 1983
            Remarks: Almost as good with
                     candies as Fliegel.
```

Name: Diana Hellespont
Address: 145 Printer Rd., Apt. 134
Compucity, USA
12349
SS#: 95-256-3973
Age: 15
Started: Feb. 12, 1983
Route: 2
Candy boxes sold: 0
Cookie boxes sold: 0
Last update: Feb. 12, 1983
Remarks: none

Name: Ruby Begonia
Address: 98 Commodore St.
Compucity, USA
12387
SS#: 25-871-4528
Age: 13
Started: Feb. 12, 1983
Route: 3
Candy boxes sold: 37
Cookie boxes sold: 21
Last update: Mar. 26, 1983
Remarks: Not doing badly for 13 yrs

Name: Frankie Gezundtheit
Address: 1131 Oracle Ave.
Compucity, USA
12311
SS#: 11-776-1755
Age: 14
Started: Feb. 12, 1983
Route: 5
Candy boxes sold: 21
Cookie boxes sold: 34
Last update: Apr. 3, 1983
Remarks: none

```
               Name: Al Emenopee
            Address: 873 Computer Ave.
                     Compucity, USA
                     12322
                SS#: 14-141-9876
                Age: 15
            Started: Feb. 12, 1983
              Route: 4
  Candy boxes sold: 45
 Cookie boxes sold: 61
        Last update: Mar. 17, 1983
            Remarks: none
```

14) After you've pressed <left arrow> to save the last record, ORACLE asks once again if you want another record. This time, instead of defaulting to 'yes'

    PRESS:n

    for 'no'.

15) The prompt changes to **Enter Exit, Add or Update.**

    PRESS:e

    for 'exit'.

The prompt line now reads: **Scratching key files.** This sounds very alarming, but it's nothing to worry about. Remember, we created two key fields when we first laid out our record format -- a key field being a field that gets automatically sorted into alphabetic or numeric order by ORACLE. To do that, ORACLE creates a special file of its own, called a **key file***, in which it sorts and stores the key field data. Whenever you modify a file in any way -- by adding, changing, or deleting records -- ORACLE has to **scratch*** (erase) the old key file so that it can re-create it with your changes. Even though we don't have any previous key files yet, ORACLE is just going through the motions to make sure.

After the key file is scratched, the prompt changes to **Please stand by...** and shortly after that to **Please stand by...Writing file...** You've got nothing to do now but wait while your disk drive whirrs away and ORACLE rewrites its key files.

When ORACLE is done, it takes you back to the **Main Menu.**

Lesson 7 - What Have You Done? - II

ON SCREEN: **Main Menu**

To see how things have changed, let's take another look at our file list.

1) Go to **Show All Files** in the **Disk Utilities** menu. (If you can't remember how, look back to **Lesson 2** or **Lesson 4.**)

   Last time we looked, we had only two files on the disk: **TREATS.REL** and **TREATS.HDR.** Now there are two more: **TREATS.KY01** and **TREATS.KY10.** The first is the key file in which ORACLE has stored the alphabetically sorted contents of field #1, the 'last name' field.

   Similarly, **TREATS.KY10** contains all the route numbers from field #10, arranged numerically.

2) Return to the **Main Menu.**

Lesson 9 - Searching for Data

ON SCREEN: Main Menu

Searching for data is one of the most important of all ORACLE functions. Search procedures give you instant access to any record or group of records in your file, even when you're not exactly sure which ones you want.

Search procedures can be used in most of ORACLE's data--management options. In this lesson, we'll look at our records using the **View Records** option, which is choice 1 under **4 = Searching and Reporting** on your **Main Menu.** Always use this option when you simply want to browse through a file, because from within **View Records** you can't change your file in any way, not even by mistake.

1) To view records

PRESS:4

When the **Search and Report** menu appears

PRESS:1

The screen displays a copy of the first page of your record format.

It is only possible to search for records using data that appears on the first page of each record.

The cursor is blinking in the first space of the first field. The prompt line reads **Enter search data (press <left arrow> when done).** There are several different search data options:

- **String Searches***, in which ORACLE looks for records containing an exact match of a specific group ('**string**'*) of letters, numbers, or other characters;
- **Wild Card Searches***, in which ORACLE accepts any character at specified points within a string search;
- **Unequal Searches***, in which ORACLE looks for records containing data that is either greater than or less than some indicated value;
- **Any Match Searches**, in which ORACLE finds all records, regardless of contents.

Let's now take up each of these search-data options in detail.

2) **String Searches**: To run a search for all records that contain a specific string of characters in a given spot in some field, you type those characters in the same position in the field in which they appear in the record.

Search strings must be typed in the correct position in the field -- otherwise, the search will not work. Also, the string must be typed exactly as it appears in the record, including all punctuation. (Capitalization, however, is an exception: when it comes to search strings, ORACLE doesn't care whether you use capitals or small letters.)

In the simplest type of string search, you specify data fully to identify a specific record. For example, let's suppose you want to find Diana Hellespont's record. We'll search for the record from the 'first name' field. To get your cursor into the second field

PRESS:RETURN

or

PRESS:<cursor down>

The cursor is now flashing in the first space of the second field. Now

TYPE: Diana

and then

PRESS:<left arrow>

The screen prompt changes to **Searching,** your disk drive whirrs for a moment, and then the first page of Diana Hellespont's record appears.

In the upper right-hand corner of the screen you'll now see something new -- **R=0007.** This is the record number of the record on the screen. ORACLE automatically assigns this number when the record is first written into the file. Generally, record numbers (R# for short) correspond to the order in which records were entered (with one small exception, which we'll encounter in the next lesson). Diana Hellespont's record was the seventh one we entered (go back and check if you don't believe it) and is therefore numbered '0007'.

At the bottom of the screen is a new prompt line: **Exit, Next, Prev, Match, or Rec.# ?** These are your options for continuing the search (if, for example, you were looking for other Dianas, or if you wanted to browse through your file). We'll investigate these options in a moment. Right now, though, we've got the record we want on the screen.

3) Once you've got the record you want, you can break out of the search routine by striking <any key> (except e,n,p,m, or r). A good habit is always to use the space bar to end the search. So

PRESS:<space>

A new prompt line appears: **Is this the record you want (y/n)? [y]** with a cursor flashing over the letter 'y' beside it. ORACLE is very considerately giving you a chance to change your mind about ending the search. If at this point you

PRESS:n

for 'no', the previous prompt line returns, and you are back in the search mode. However, we have the record we want, so

PRESS:<space>

again. 'Yes' is the default response to this prompt, so to tell ORACLE that the search is over,

PRESS:RETURN

4) A new prompt appears: **Do you want the next Page? [y]** Any response to this prompt will cause the first page to disappear from the screen, and you won't be able to get it back without going through the whole search procedure again. So when you get to this point in a search, be sure you are done with the page on the screen before you go on.

From here on several things can happen. If you (Don't do it!) **PRESS:**n for 'no', ORACLE will return you to the **Search and Report** menu. If you

PRESS:RETURN

for 'yes', the second page of the record appears. It's empty because we didn't enter any remarks for Diana. The prompt is the same: **Do you want the next page? [y]** This may seem strange, since there is no next page in this record. However, at this point ORACLE doesn't know that. Where there is no next page, answering either 'yes' or 'no' will take you back to the menu. So

PRESS:n

or

PRESS:RETURN

Either way, your search, in which you entered the exact string necessary to find a record, is over.

5) You can also use strings to search for groups of records. For example, to find all the people in your file whose first names begin with the letters 'Al', get the **View Records** screen back, move the cursor to the first spot in the second field, and

TYPE: Al
PRESS:<left arrow>

In a moment R# 0003 appears, Allan Fliegel. 'Allan' certainly begins with 'Al', but it's not the only first name in our file that does. To find these other records, you use the 'continue search options' displayed in the prompt. Each option gives you a different way of continuing your search.

6) The **Next** option gives you the next record in order of record number. Allan Fliegel's record is R# 0003. If you now

PRESS:n

R# 0004, Annie Ablative, will appear on the screen. If you

PRESS:n

again, R# 0005, Alicia DeSoto, will appear, and so on, all the way through to the final record of the file. Try it. Keep pressing 'n' until you have R# 0010 on the screen. If you now press 'n' again, something new will happen.

PRESS:n

Since there is no next R# in this file, a new prompt appears: **No such record - press any key.** So

PRESS:<any key>

After a slight pause, ORACLE loops back to R# 0001, Maria California. As long as you keep striking 'n', ORACLE will keep cycling you through your file, taking you from the last R# back to the first, each time you reach the end. If you now

PRESS:n

and then

PRESS:n

again, you'll find yourself back where you started with R# 0003.

7) Just as the **Next** option gives you the next R#, the **Prev** option gives you the previous R#. However, there is one small difference between the two. When there is no previous R# (because you've arrived at R#0001), striking 'p' for 'previous' simply does nothing. Try it.

8) If you've pressed 'p' two or more times, you should have R#0001 on the screen. At this point, you could use the **Next** option twice to get back to R# 0003. But there's another way of going about it -- the **Rec.#** option. And it's a lot faster when you are working in a very large file with many R#s. With the **Rec.#** option, you can jump to any record instantly as long as you know its number. So, to get back to R# 0003

PRESS:r

for **Rec.#**. The prompt changes to **Enter record number.** So

ENTER:3

(you don't have to type all the leading zeros) and ORACLE takes you straight to R# 0003. You can jump to any record in your file using the **Rec.#** option. Try it. Then go back to R# 0003, where we started out.

None of the 'continue search' options we've explored so far have referred to our original search string -- the letters 'Al' in the first two spaces of our second field. After Allan Fliegel's record appeared, we used 'n', 'p', and 'r' to move to other records: 'n' displayed the record with the next highest R#, regardless of whether the first two characters in the search field were 'Al'; 'p' displayed the record with the next lowest R#, regardless of whether the first two characters in the search field were 'Al'; and 'r' gave us any R# we wanted, regardless of whether the first two characters in the field were 'Al'.

9) To continue a search according to your original search string, you use the **Match** search option.

PRESS:m

for 'match'. ORACLE displays R# 0005, Alicia DeSoto, whose first name also begins with 'Al'.

PRESS:m

again and you get R# 0006, Albert Galapagos, ditto.

PRESS:m

again and you get R# 0010, Al Emenopee, the last record in our file and therefore the last match you will find. If you

PRESS:m

one more time, something new happens. A new prompt appears: **End of file (press any key.)** Whenever you reach the last match in your file, this prompt appears -- whether or not the last match is also the last record, as it happens to be in our case. If you now

PRESS:<any key>

ORACLE takes you back to the **Search and Report** menu.

10) You can run searches for strings of numbers just as easily. Just to prove it to yourself, run a search to find all the 15-year-old volunteers on our list. You should find two of them. (Hint: type '15' into the 'age' field and then search for matches.)

As our example has demonstrated, if you interrupt a string search with any of the other 'continue search' options - **Next, Prev,** or **Rec.#** - ORACLE still remembers the original search string, and you can go back to looking for matches whenever you want. If you press 'm' again, ORACLE will display the match that follows the last one it found.

11) **Wild Card Searches:** Just as a 'wild card' in poker can represent any card in the deck, a 'wild card' in an ORACLE search string can represent any character in a field. In a wild card search you embed one or more '?' symbols in your search string. ORACLE understands the '?' in a search string to mean 'any character'. It will find all records that match your specific search string, and that contain any other character where you have entered '?'.

To make this clearer, let's run a wild card search ourselves. Get the **View Records** screen back, and take the cursor down to the second field. Now, for search data

TYPE:??b

You're asking ORACLE to find all records in which the third letter in the first name is 'b'. Now

PRESS:<left arrow>

The first record displayed is R# 0006, Albert Galapagos, whose first name has a 'b' in the third position.

PRESS:m

for match and you get R# 0008, Ruby Begonia, whose first name also has a 'b' as its third letter.

PRESS:m

once more and you get the **End of File** prompt, indicating that there are no more matching records left in the file.

PRESS:<any key>

to get back to the menu.

The wild card symbol can be embedded anywhere in a search string. As with normal string searches, wild card searches work with both alphabetic and numeric strings, and give you matching records in order of R#.

12) For a more realistic wild card search, we might ask ORACLE to find all records updated in April - in other words, to find records containing '04' in the third and fourth spaces of the 'last update' field. Go to **View Records**, move your cursor down to the last field on the page, and

TYPE:??04

Now

PRESS:<left arrow>

The first record to come up is R# 0003, last updated April 1, 1983.

PRESS:m

The second record to come up is R# 0009, last updated April 3, 1983.

PRESS:m

and you get the end of file prompt. There are no more records in the file that were updated in April.

PRESS:<any key>

to get back to the menu.

13) **Unequal Match Searches**: Unequal match searches allow you to search for records that contain ranges of values in a given field. In our volunteer file, for example, if you wanted to find all the volunteers older than 15, you would use an unequal match search. In the same way, to find all the volunteers whose first name begins with a letter that comes after 'D' in the alphabet, or that comes before 'J', you would use unequal match searches.

To see how this kind of search works, let's find all the volunteers older than 15. Go to **View Records**, and move the cursor to field #4.

PRESS:>

That is, hold down the shift key, and press the '>' key. Nothing happens in the field itself, but a '>' symbol appears on the top line of your screen. Now

TYPE:15

in the field. The search string is now complete.

PRESS:<left arrow>

The first record to come up is R# 0003, Allan Fliegel, aged 18. Now use the **Match** option to find all the other matching records. There are three more. Notice that this search does not produce 15-year-olds.

14) Now, use a '<' search on your own to find all the volunteers who have sold fewer than 20 boxes of candy. There are three of them.

15) Okay, now let's find all the volunteers whose first name starts with a letter that comes after 'D' in the alphabet. Get your **View Records** screen back, and move your cursor to the first position in the second field.

    PRESS:>
    TYPE:d
    PRESS:<left arrow>

    You should come up with four matches.

For unequal matches to work, all the original data in your records must be entered properly -- words left-justified and numbers right-justified. Similarly, all your search data must be entered properly -- word search strings left-justified and number search strings right-justified.

16) **Any Match Searches**: If you're just browsing through your file without any particular record in mind, or if you want to look at the entire file, you can use an any match search. To do this you type an '*' in the first position of any field and then run the search. ORACLE will then give you all the records in your file in order of record number.

If, for example, you type an '*' in your 'apartment #' field and then run the search, you'll see that ORACLE displays every record from 0001 to 0010 when you step through with the 'm' key.

In '*' searches, the 'm' and 'n' keys produce the same result. The only difference comes at the last record of the file. If you strike 'n' at the last record, you cycle forward to the first record in the file; but if you strike 'm' at the last record, you must exit to the **Search and Report** menu.

17) **Searching in Key Fields**: You may have noticed that we have avoided running any searches in our key fields - the 'last name' field and the 'route #' field. That's because searches in key fields work a little differently than searches in non-key fields.

Key fields are different from non-key fields in that they are automatically sorted, either alphabetically or numerically, depending on the kind of data in the field. As a result, when you search in a key field, ORACLE displays your matching records in sorted order.

To demonstrate key field searching, go to **View Records**, and move the cursor down to the 'route #' field.

Now run an '*' search. The first record displayed is route # 1, but R# 0003. Now

PRESS:m

The second record is route # 2, R# 0007. If you keep pressing 'm', you will get all the records in order of route number, from 1 to 10, with their R#s all mixed up.

18) Now run a '*' search in field #1. By stepping through the search with 'm' or 'n', you will get all your records ordered alphabetically by last name.

Key field searches always present records according to the way they are sorted in the key field. This is true in any kind of key field search.

There are two prohibitions which apply to searching in key fields:

- Wild Card searches will not work properly in key fields.
- Unequal Match searches will produce an **End of file** message when the search string has an exact match in your file.

Further Explorations

We've just spent a long time on searches, yet we've hardly scratched the surface. We've performed only the simplest kinds of searches. We'll look at much more complex searches in the **ORACLE Design** section of the manual. In the meantime, however, you'll find it useful to make up some more searches by yourself, and run them just for practice. You might, for example, try to figure out how to use a wild card search to find volunteers who have sold between 60 and 69 boxes of cookies. Or you might try to list in alphabetical order all the volunteers aged 15 and older. To a large extent, ORACLE is just as powerful as your ability to design effective, efficient searches. The only way you can learn to do that is with practice.

Lesson 10 - Modifying Existing Files

ON SCREEN: **Main Menu**

To keep your files current, you'll often want to change some of the data in an existing record. In our sample file, for instance, a volunteer might move, and we'd have to make a change of address. When volunteers report in with new sales, we'd want to update their sales figures. Or, to take a more extreme example, a volunteer might quit entirely, and we'd have to delete that record from the file and create a new record for a replacement. ORACLE allows you to make such changes quickly and easily.

Note that these are changes to the data contained in a record -- not to the record format itself. We'll deal with changing record formats later. In this lesson we are concerned only with changing data, not with changing formats.

1) The **Main Menu** option for modifying files is **3 = Modify an Existing File.** So to begin

   PRESS:3

   The is the same Main Menu option we used to add the original data to our files. Again, the **File Maintenance** screen appears, containing an empty record format and the prompt: **Enter Exit, Add, or Update. Exit** takes you back to the Main Menu. **Add,** as we have already seen, allows you to add new records to your file. The choice we want now is **Update.** So

   PRESS:u

2) The top line of the screen now reads **UPDATE RECORD.** The cursor is blinking in the first space of the first field, and the prompt line reads: **Enter search data (press <left arrow> when done).** Why a search prompt? Well, if you want to change a record, you have to find it. As we said at the beginning of the last lesson, searches are used in many ORACLE procedures.

   Let's suppose we have an address to change. The California twins have moved from their old address on Commodore St. to 1000 Dual Dr., Apt. #302. Their zip is now 12386. To get their records on the screen you have to run a search. So, into the first field

TYPE: Cal

then

PRESS:<left arrow>

Your screen will display either R# 0001, Maria California, or R# 0002, Jose California. At the bottom of your screen is a new prompt: **Change, Del, Next, Prev, Match, Rec.#, Exit?.** We want to change the record on screen, so

PRESS:c

for 'change'. The cursor moves to the first field and a new prompt line appears: **Enter changes (press <left arrow> when done).**

3) To make the changes, take the cursor down to the 'street' field, and

TYPE:1000 Dual Dr. <space,space,space>

to overtype the old street address with the new. The spaces simply erase the last few characters of the old data. Now

PRESS:<cursor down>

to move the cursor to the next field, and

TYPE:<space>302

to overtype the old apartment number and enter the new one. Finally, move the cursor down two fields to 'code' (the city stays the same), and

TYPE:12386

to overtype the old code. All the changes are now entered so

PRESS:<left arrow>

The second page of the record appears. If there were any changes here, we could now make them. Since there are no changes,

PRESS:<left arrow>

A new prompt appears in reverse video: **Writing Record.** ORACLE is now writing the changes to the disk. When the prompt disappears, ORACLE returns to the first page of the record we just changed.

4) However, we're not done yet. We have to make the same change of address to a second record. To get the other 'Cal' record

   PRESS:m

   And the record for the other California twin appears. You can now make the same series of changes in this record as you did in the last.

   When ORACLE has written your changes to the disk, the **Update** prompt reappears. There are no more changes to be made so

   PRESS:e

   for 'exit'. ORACLE returns to the **File Maintenance** screen.

5) There are now several options. If you had completed all the changes you wanted to make in your file, you would exit. However, we're not quite done yet. We want to delete a record (because a volunteer has left us) and then add another (because we've got a new volunteer ready and waiting) to take its place. First we'll make the deletion. We do that in the **Update** mode. So

   PRESS:u

   to get the **Update** screen back. The volunteer we want to delete is Diana Hellespont. So run a search to get her record onto the screen.

6) You've got R# 0007 on the screen. Diana, as you can see, has sold no cookies, no candies, and has not reported in since she took on the job. We're not exactly heartbroken to see her go. To delete her record

   PRESS:d

   for 'delete'. A new prompt appears: **Deletion - are you sure? [n]** with the cursor flashing over the 'n'. ORACLE is giving you a chance to think twice about deleting a whole record all at once. If at this point you change your mind,

PRESS:RETURN

to default to 'no', and you go back to the **Update** prompt. However, we do want to delete this record, so

PRESS:d

again. Once more the **Are you sure?** prompt appears. Now

PRESS:y

for 'yes'. Your disk drive whirrs. After a moment ORACLE takes you back to the **Update** screen. Our file no longer contains R# 0007, and Diana Hellespont is no longer among our circle of friends. On the prompt line are only the search options. Since we have no more updates to make,

PRESS:e

for 'exit'. ORACLE returns to the **File Maintenance** screen.

7) But we're still not finished. We want to add a new record to replace the one we've just deleted. As when we first added data to our file, we do that in the **Add** mode.

PRESS:a

for 'add'. When the prompt changes,

PRESS:RETURN

to default to the keyboard. You can now enter the data for the new record, just as you did earlier. Here's the information:

Name: Betty Iguana
Address: 19 Oracle Dr., Apt. #1
Compucity, USA
12333
SS#: 71-319-8886
Age: 15
Started: May 14, 1983
Route: 2
Candy boxes sold: 0
Cookie boxes sold: 0
Last update: May 14, 1983
Remarks: none

Notice that we've given Betty the route that Diana had. Because she's just begun as a volunteer, she has a zero sales record, and her last update date is the same as her starting date. Let's hope she does better than her predecessor.

Once you've added all the data and refused another record, ORACLE returns to the **File Maintenance** screen.

8) All our changes are now complete, so

PRESS:e

for 'exit'. The prompt changes to **Scratching Key files.** Since you've changed the data in your file, the key files that ORACLE wrote when you first created your file are no longer accurate. Each time you exit from the **File Maintenance** mode, ORACLE erases and rewrites all its original key files. When the old files are scratched, a **Please standby** message appears. Shortly after, ORACLE announces that it is **sorting.** (Chances are you won't even see this message because, on small files like ours, the sort takes only a fraction of a second.) After the sort, ORACLE displays **writing file.** When the message disappears, our first key file has been rewritten. In a moment, the **writing file** prompt reappears, and the second key file is rewritten. ORACLE then returns to the **Main Menu.**

Before we leave the subject of modifying files, there are a few general remarks to be made.

ORACLE allows you to leave the **File Maintenance** mode with **PRESS:**<escape> -- f8 on the Commodore 64 and (shift)RUN/STOP on PET/CBMs. However, when you escape from the mode in this way key files are not rewritten. So if you've made changes in key fields, your key files may no longer be accurate. The moral is: Always exit from the **File Maintenance** mode by using the **Exit** option. If you make that your invariable rule you'll never make a mistake.

Remember also that key files are not rewritten until you **Exit** the **File Maintenance** mode. Before exiting, however, you might want to make many different kinds of changes, in many different records, involving a variety of searches. But once you've made a change in a key field, your key files are no longer accurate. Any new searches you run to look for other records may get confused. To avoid the problem, once you've made any changes to a key field, don't run any new searches in that field until the key file has been rewritten.

Finally, in the last lesson we said that generally R#s are assigned to records in the order in which they are written. Well, here's the exception. By deleting Diana Hellespont's record, we left the place for R# 0007 blank. When we wrote in the new record, it was assigned R# 0007 to fill that blank. ORACLE will always fill any blanks with newly added records before it starts using new R#s. If you don't believe it, go to **View Records** and check.

Lesson 11 - Setting Up the Printer [P]

ON SCREEN: Main Menu

So far we've just looked at information on the ORACLE screen. But one of ORACLE's most powerful features is that it lets you print out **hard copy*** based on your data. Setting up your printer to print from ORACLE is a fairly simple matter.

1) First, make sure your printer is properly connected to your system. Consult your printer manual for directions.

2) The printer set-up option is in the **Search and Report** menu, so to begin

   PRESS:4

   and then

   PRESS:5

3) A new screen appears entitled **Printer Setup.** The prompt reads: **Enter number of linefeeds [0].** If your printer has an automatic **linefeed*** (consult its manual),

   PRESS:RETURN

   to default to 0 (zero). If your printer does not have an automatic linefeed,

   ENTER:1

   Either way, you've set your printer to type your reports single-spaced.

4) A new prompt appears: **Printer device number [4].** The default value is 4. Most printers are assigned device number 4. On some systems the printer may be device 5 or 6 (consult your manual). If your printer device is 4,

   PRESS:RETURN

   for the default value. If not

   ENTER:<correct value>

5) Another prompt appears: **Output? (Screen or Printer) [s]**. The default value is 'screen'. You use the default to shut off the printer. To set it up

PRESS:p

6) A fourth prompt appears: **Enter page header* or RETURN.** This option allows you to enter a line of text to be printed at the top of each report page. For now, just

PRESS:RETURN

to default to 'no header'.

7) A fifth prompt appears: **Lines/page? [55].** ORACLE wants to know how many lines you want to print per page. The default is 55, which is a good number to use on standard 66-line pages.

PRESS:RETURN

to default to '55'.

8) A sixth prompt asks: **Ascii or Cbm Code?** If you're using a Commodore printer,

PRESS:c

If not,

PRESS:a

9) ORACLE returns to the **Search and Report** menu. Your printer is now set up to print single-spaced pages of 55 lines each. Let's run a few test printouts.

ORACLE will print any screen with a flashing cursor in a field. So, to get a printed copy of the first page of your record format,

PRESS:1

to get into the **View Records** mode. When the empty record format appears with the cursor flashing in the first field,

PRESS:"

and your printer will print an exact copy of the format, using square brackets to represent field markers.

10) Suppose you want a printout of an actual record? You should still be in **View Records,** so run a search to get Annie Ablative's record on the screen. Now

PRESS:<any key>

The prompt changes to: **Is this the record you want (y/n)? [y].**

PRESS:RETURN

to default to 'yes'. The prompt changes to **Print this Screen? [n].** The default here is 'no', so to print the screen,

PRESS:y

Your printer immediately prints the first page of Annie Ablative's record, and ORACLE asks if you want the next page. If you answer 'yes', you go through the same routine to get a printout of page 2. If you answer 'no', you end up back at the menu.

11) You can also get a printout of your field list. This will be a very handy item later on, so

PRESS:6

to list fields. If your printer is on, ORACLE automatically prints the list as it comes up. When the printout is done

PRESS:<any key>

to get back to the **Search and Report** menu. We'll look at printing actual reports in the following lessons.

Lesson 12 - Designing Reports [P]

ON SCREEN: Search and Report menu

ORACLE can also give you printouts that don't merely repeat the recorded data but arrange it in new ways or perform calculations with it. The option that allows you to design such special reports is called, sensibly enough, **Design a Report.** In this lesson, we are going to introduce some of the basic features of report design. But first, let's shut the printer off for the time being. To do that

PRESS:5

to go back to the **Printer Setup** screen. Now

PRESS:RETURN
PRESS:RETURN
PRESS:RETURN

and you're back in the **Search and Report** menu. The printer is now shut off, and we're ready to begin.

1) To enter the **Design Reports** mode

   PRESS:2

   A blank screen comes up with the prompt: **Use a saved format? [y].** The default is 'yes'. The 'format' referred to in the prompt is not the record format but a special **report format***. Since we don't have any format saved yet,

   PRESS:n

   for 'no'. The screen changes; the cursor is blinking in the upper left-hand corner, and the prompt reads: **Enter format (<up arrow>=start field, <left arrow>=quit).** Now we can create our first report format.

2) We'll start with a very simple report that will give us the name of each volunteer and the number of boxes of candy each has sold. The names will be listed one above the other on the printed page, and the number sold will be beside each name. We'll label each column at the top of the page, and we'll print a line at the bottom to name the author of the report.

To begin we'll establish which data fields we want to print. Move your cursor down to the beginning of the fourth screen line (we'll come back to the top lines later), and

PRESS:<up arrow>

to start a field, just as you would in creating a record format. A new prompt appears: **Enter field # for position.** Each time you create a field in this mode you must give it a number. We want the 'first name' field from our original record layout -- that is, field #2. (Here's where a printout of your field list comes in handy.) So

ENTER:2

Another prompt appears: **02=first name.. len=20 This one O.K.? [y]** The default is 'yes'. ORACLE is simply asking if this is the field you want, and if its current length (20) is okay for this report. Try pressing 'n' for 'no'. ORACLE changes the prompt to: **03=SS#......... len=11 This one O.K.? [y].** As long as you keep striking 'n', the prompt will move on to the next field in your field list. When you reach the end of the list, the prompt loops back to field #1, and then keeps on going. Keep pressing 'n' until you come back to field #2. Now

PRESS:RETURN

to answer 'yes'. A close field marker appears with the letter 'b' after it. ORACLE assigns each of your data fields a symbol for its own reference. (For an explanation of how these labels are assigned, see the reference section of the manual.)

3) On this same line we also want to print the number of boxes of cookies each of the volunteers has sold. Space over 5 spaces and

PRESS:<up arrow>

to start the field. 'Cookies' is field #12 in our record format, so

ENTER:12
PRESS:RETURN

The close field marker appears with the letter '1' beside it. Our second data field is now done. However, to get a neater printout, we'll push it over to the right of the screen so that our numbers will be printed down the right edge of the printout. To move the field over, move your cursor to the space between the two data fields. Now, use (shift)INST/DEL to INSerT spaces between the two fields, and push the '1' to the edge of the screen. If you go too far, use DEL to pull it back. All screen editing functions work quite normally on the **Design Reports** screen.

4) We've specified the data we want to print. Now, we'll create the column labels for the top of the report. Home your cursor to the upper left-hand corner of the screen, and

PRESS:<up arrow>

to start a new field. The **Enter field #** prompt appears. This time we don't want to print data; we want to print descriptive text. Report fields for printing descriptive text are called **variable fields***. And, like data fields, they are identified by a number. Variable fields for printing text at the top of a report page are identified as '**field type*** 101'. So

ENTER:101

The prompt now reads: **Enter variable field data and return.** This field is going label our 'name' column, so

TYPE:NAME

into the field and

PRESS:RETURN

The field closes, and a funny-looking symbol appears after the close field marker. This symbol identifies all 101 fields.

5) The **Enter format** prompt is back and we are ready to create the label for our second column. First, use the space bar or <cursor left> key to move the cursor over 17 spaces. Now, create the next field exactly as before, but this time type in 'COOKIES SOLD' when ORACLE asks for the variable field data.

Here are the steps:

PRESS:<up arrow>
ENTER:101
TYPE:COOKIES SOLD
PRESS:RETURN

Again, the field closes with the same funny symbol. The top line of the report is now complete.

6) Now we'll create a line to separate our column labels from the body of the report. Move the cursor down to the next line with

PRESS:RETURN

To draw our line, we'll use a 101 field, because once more it will only appear at the beginning of the report page.

PRESS:<up arrow>
ENTER:101

and

PRESS:<hyphen>

36 times. This will take you close to the right edge of a 40-column screen, with room left for the field close marker and the funny symbol. If you're using an 80-column screen, you are certainly not restricted to the left half of the screen to design reports. However, everything in this manual is designed to work with either size screen. Now

PRESS:RETURN

to close the field, and

PRESS:RETURN

again to go down to the next line.

7) Now, on our printout we want to separate the top-of-page information from the body of the report with a blank line. To create a blank line on the printed page you use an empty field (the printer ignores all blank screen lines). So

PRESS:<up arrow>
ENTER:101
TYPE:<space>
PRESS:RETURN

The top of the screen is now completed.

8) The next step is to create variable fields for the text at the end of the report. First, we'll create a blank line between the body of the report and our end-of-report. Move your cursor down to the line below your data fields. Make sure it's at the left edge of the screen. We need an empty field to create a blank line in the printout, so

PRESS:<up arrow>

to start the field. End-of-report variable fields are numbered 102, so

ENTER:102
TYPE:<space>
PRESS:RETURN

Notice that 102 fields are followed by a different kind of symbol.

9) Now we'll create a line across the bottom of the report, just as we did across the top. The procedure is exactly the same, only the field type is different -- 102 instead of 101. Make sure your cursor is at the beginning of the next line.

PRESS:<up arrow>
ENTER:102
PRESS:<hyphen>

36 times, and

PRESS:RETURN

10) One more field finishes our report. Make sure your cursor is at the beginning of the next line.

PRESS:<up arrow>
ENTER:102
TYPE: Report prepared by:

followed by your own name. (Use only one of your names if you don't have room for the whole thing.)

PRESS:RETURN

to finish the field.

11) Our report screen is now complete, so

PRESS:<left arrow>

to end the creation process. The field symbols disappear, and a new prompt appears: **Any Changes? [n].** Now is the time to look over your screen to see if everything is as you want it. Is everything lined up okay? Spelling correct? If you want to change anything strike 'y' for 'yes', and you'll find yourself back in the creation mode. If you're satisfied with your screen,

PRESS:RETURN

to answer 'no'. A new prompt appears: **Save the Format? [y].** We want to save this format for later use, so

PRESS:RETURN

to default to 'yes'. ORACLE now asks you to **Enter format number.** Every time you save a format you have to give it a single-character identification code (letters are okay too).

PRESS:1

to label this format #1. After a moment, the familiar **Enter search data** prompt appears, and the cursor is flashing in the first data field on the screen.

12) Setting up a search in this mode is the same as in **View Records.**

TYPE:*

in the 'first name' field, and then

PRESS:<left arrow>

to run an any match search. After a few moments our first record appears -- Maria, who has sold 23 boxes of cookies.

The prompt for continuing the search is a little different from the one we encountered in **View Records.** This prompt reads: **Enter q(quit), n(next match), c(cont.).** Typing 'q' aborts the search and takes you back to the **Search and Report** menu. The **next match** option gives you the next record that matches the search data. For example, if you now

PRESS:n

onto the screen comes Jose, who has sold one dismal box of cookies.

PRESS:n

again, and up comes Allan, who has done somewhat better, at 67 boxes.

The **cont.** option calls up all matching records, one after the other, without your having to strike 'n' after each one. Keep your eye on the screen and

PRESS:c

for 'continuous'. The appropriate data from all the remaining records in this file come up in turn. The **cont.** option lets you print reports without having to hang over the keyboard hitting 'n' all the time. Remember, though, you can always pause in the 'continuous' option by pressing <any key> while the search is running. We'll demonstrate this when we actually print the report.

When the search is over you get the **End of file** prompt. <Any key> now takes you back to the **Search and Report** menu.

13) Now we're going to see what happens when we start printing with the format we've just created.

Go to the **Setup Printer** option, and set up your printer just as you did before in **Lesson 11,** with one small exception. When you come to the fourth prompt: **Enter page header or RETURN,**

ENTER: Design Reports

This page header is important. Not only is it useful for identifying your reports, but it is also necessary in order for your top-of-page lines to print properly.

14) Once your printer is set up and you're back in the **Search and Report** menu

PRESS:2

to get back into the **Design Reports** mode. This is the mode from which you print reports as well. When the blank screen appears ORACLE asks: **Use a saved Format? [y]**. We've saved our format, so

PRESS:RETURN

to answer 'yes'. ORACLE now asks for the format number.

PRESS:1

In a moment our format appears, just as we set it up. ORACLE asks: **Any Changes? [n]**.

PRESS:RETURN

to answer 'no'. Now, the **Enter search data** prompt appears, with the cursor flashing in the first data field. As before, we'll run a * search in this field.

PRESS:*
PRESS:<left arrow>

to run the search.

15) As soon as you hit that <left arrow>, your printer should chug into action. First, it prints the header: Design Reports. Then it prints the top-of-page fields: 'NAME' at the beginning of the line, and 'COOKIES SOLD' at the end. Below them comes a line of hyphens. After skipping a line, it prints our first line of data: 'Maria', and '0023', just as you see it on the screen.

PRESS:n

After a slight pause, the data from the next record appears on screen and prints directly below the first line of data on your page. If you now

PRESS:c

the printer will print all the remaining entries from your file, one below the other, in the same order as they appear on the screen. To demonstrate ORACLE's ability to pause in the middle of a 'continuous' search, let one or two more lines print out. Then

PRESS:<space bar>

The printer stops. The screen stops. And a new prompt appears: **Interrupt? [y].** Pressing RETURN ends the search and takes you back to the search prompt. Pressing 'n' for 'no' continues the search and the printout.

PRESS:n

The search and the printout resume.

After the last record is printed, the printer will again skip a line, type another line of hyphens, and print a final line that says -- 'Report prepared by: You know who'.

The screen now displays the **End of File** message. <Any key> takes you back to the **Search and Report** menu.

16) Try another search on your own. Go back to **Design Reports,** call up format #1, and ask ORACLE to print a list of all the volunteers who have sold more than 50 boxes of cookies. Use the RETURN key to get your cursor into the second data field, and run a >50 search. (Remember to enter your search data properly, right-justified with zeros.)

You should come up with a printed list of three volunteers: Allan with 67 boxes sold, Alicia with 67, and Al with 61.

When your printout is finished return to the **Search and Report** menu.

So far, we've used three types of report fields: ordinary data fields, numbered as they are in your field list; 101-type variable fields, which appear only at the beginning of each report page and which contain text that you type in; and 102-type variable fields, which appear only at the end of reports.

Now we'll take a quick look at some other types of fields.

17) Get format #1 back onto the screen. When ORACLE asks: **Any changes? [n]**

PRESS:y

for 'yes'. The field symbols appear, the cursor is on the screen, and the **Enter format** prompt is displayed.

In this example we'll explore the use of 0-type fields. Like 101 and 102 fields, 0 (zero) fields are variable fields. But unlike 101 and 102 fields, 0 fields will print anywhere within the body of the report.

We'll use the same information we did in format #1, but this time let's arrange it a bit differently. To start, we'll retype the top line. Home your cursor to the upper left-hand corner of the screen.

PRESS:<up arrow>

to start a new field, and

ENTER:101

for a top-of-page variable field.

ENTER:SALES SHEET: COOKIES

Now strike the space bar enough times to erase the remainder of the top line. (Make sure you get that close field marker at the end of the screen!) Skip over the next two lines, bringing the cursor down to the fourth line, where our data fields are. Now

PRESS:<left arrow>

to start a new field and

ENTER:0

to specify a 0-type (zero-type) variable field. Now,

TYPE:NAME:
PRESS:RETURN

Notice that the symbol following the close field marker of 0-type fields is different again. Use the space bar to erase all the remaining characters, making sure you don't go down into the next line.

Now bring the cursor back to 3 or 4 spaces from the end of the 0-type field. To add our data field ('first name') to this line

PRESS:<up arrow>
ENTER:2

In response to the **OK?** prompt,

PRESS:RETURN

to close the field. Bring the cursor back to the space between the two fields, and hit (shift)INST/DEL enough times to line up the 'b' symbol with the other close field markers at the right of the screen.

Now, we want to insert a line on which to add a couple of fields. You do this the same way you did it in the **Create Record Format** mode. Get your cursor down to the next line and

PRESS:RUN/STOP

The prompt reads: **(+)=INST, (-)=DEL LINE.** We want to insert a line, so

PRESS:+

There's now a blank line under the 'NAME' line.

PRESS:<any key>

(except, of course, the '+' or '-' keys) to shut off the insert/delete option.

To start creating our new line, which will now be the 'COOKIES SOLD:' line,

PRESS:<up arrow>
ENTER:0
TYPE:COOKIES SOLD:
PRESS:RETURN

To put the data field into this line

PRESS:<up arrow>
ENTER:12
PRESS:RETURN

Make sure that the start field markers for the two data fields line up, one above the other. Move the cursor down so we can insert another line.

PRESS:RUN/STOP
PRESS:+
PRESS:<any key>

You should now have a new line inserted, with the cursor in its first space. We now want to create an empty 0-type field to tell the printer to print a blank line between each of our records.

PRESS:<up arrow>
ENTER:0
PRESS:<space>
PRESS:RETURN

to finish the field.

18) Now we're going to do something different. We want our report data to be printed by order of route #, but we're not interested in having the route # itself appear in the printout.

Whenever you want to run a search in a data field that you don't want to appear in a printout, you add 128 to the data field number, and enter the result when ORACLE asks you to **Enter field # for position.** Adding 128 to any data field tells ORACLE not to print that field. No matter where you put this field in your format - at the top, at the bottom, or even inserted into the body of the format among other fields - the printer will ignore it.

Move your cursor down to the first space in the next line, and, once again, insert a blank line. (You know how.)

From your field list you know that the data field number for 'route' is 10. 10 + 128 = 138, so,

PRESS:<up arrow>
ENTER:138
PRESS:RETURN

A reverse video character will appear after the field -- as after all non-printing fields.

Your new format is now completed.

PRESS:<left arrow>

If there are any changes, go back and make them now. If everything's okay,

PRESS:RETURN

for 'no'. We want to save this format, so

PRESS:RETURN

again, for 'yes'. Now,

PRESS:2

to give the format a different number from our last one.

19) After saving the new format, ORACLE presents the **Enter search data** prompt. Press RETURN twice to move your cursor to the 'route #' field.

PRESS:*
PRESS:<left arrow>

The first volunteer to appear is Allan Fliegal, route #1. (Remember, 'route #' is a key field, so that our * search will give us back all the records in order of route #, from 1 to 10.) As the data comes up on the screen, the report starts to print. To continue the report,

PRESS:c

Take a look at this new printout. The printer has ignored the 138 field entirely; it hasn't even left a blank line in its place. Notice also that each two-line block of data is separated from the next by a blank line. This blank line was created by the empty 0-type field we inserted into the body of the report.

20) Now, we'll look at another feature of variable fields. Here's what to do.

Get format #2 back on the screen. Tell ORACLE that you want to make some changes. Take the cursor down to the beginning of the 138 field. Hit <up arrow> to start a new field, and specify a 0-type field. When ORACLE asks you to enter variable field data,

TYPE:?

followed by 20 blank spaces. Close the field with RETURN, and press <left arrow> to tell ORACLE you're done. You don't have to bother saving this format.

By typing the '?' as the first character of a variable field, you've created a format that allows you to add new variable text (up to 20 characters, in this example) to your report even as it's being printed.

Let's try it. Run an any match search in the name field. When the data fields from the first record are printed, your printer should pause, and the cursor should be flashing over the question mark. The prompt reads: **Enter variable field data.**

Type whatever you like onto the screen into the '?' field. When you hit RETURN, your printer will print what you've just typed, and the continue search prompt will appear.

PRESS:c

The printer will now print continuously, pausing at the same field in each record to allow you to type in new information.

In this lesson we've looked at the basics of report design. We've seen some of the fundamental field types, and we've looked at how to create fields for variable report data. You've got the skills now for creating all kinds of single-page reports. But ORACLE is actually capable of much more. For full information on the design of more complex, multi-page reports, see **ORACLE Design**.

Lesson 13 - Special Sorts [P]

ON SCREEN: Search and Report Menu

When we created our record format, we created two key fields: the 'last name' field and the 'route #' field. Remember, ORACLE automatically sorts key fields, either alphabetically or numerically, depending on which kind of data they contain.

But what if, some way down the road, you want to sort data contained in a non-key field? Do you have to go back and recreate your record format just for this one job?

No, you don't. ORACLE gives you the ability to sort any field at any time, using the **Sort Fields** option of the **Search and Report** menu. Sorts created in this way are called special sorts. As you'll see, they are a lot more powerful and versatile than ordinary sorts. In this lesson we're going to learn how to create and use special sorts.

Before we actually start sorting fields, we'll create a new report format for printing out the results of our sorting experiments. The format is very simple, and you should be experienced enough by now to set it up on your own.

First, make sure your printer is on and set to print 55 lines per page, with a header that reads: "Sort Examples".

Here are the specifications for our report format.

The report is to be 4 lines long. Line one contains our 'first name' field (field #2). Line two contains our 'last name' field (field #1.) Line three contains our 'age' field (field #4.) And line four contains an empty 0-type field (to force a blank line between blocks of data in the printout). To help you remember which field is which, you can simply type a brief out-of-field description of each field after its close field marker. As long as you haven't placed them within any kind of field, these labels won't affect your printout in any way. Once you've created the format, save it as format #3.

Before we start sorting, let's get an unsorted printout, just for comparison. Use an * search in either the 'first name' field or the 'age' field to do this. Because neither of these is a key field, they'll produce exactly the same result: the data will appear in order of R#.

Now we can start sorting.

1) Get the **Search and Report** menu on the screen. **Sort Fields** is menu item **7**, so

   PRESS:7

2) A new screen appears with the single prompt: **Enter field number to sort [1].** The default is '1'. We want to sort field #2, so

   ENTER:2

   A new prompt appears: **Enter sort depth (15 max) - [5].** The sort depth is the number of characters (reading left to right) in the field that ORACLE takes into consideration when doing the sort. For example, if you sorted an alpha field to the depth of 1, ORACLE would give you all the 'A's grouped together, then all the 'B's, then 'C's, and so on. But since ORACLE hasn't looked at the second character, your list of 'A's wouldn't be alphabetized, nor your 'B's, nor your 'C's. If you sorted the same field to a depth of 2, you'd get all the 'AA's grouped together, then the 'AB's, then the 'AC's, and so on. The maximum depth to which you can sort a field is 15. For our example, we'll use a sort depth of 2, so

   ENTER:2

   At the bottom of the screen, a new prompt: **Subsort req'd?** appears. We'll look at subsorts in a moment. For now, default to 'no' with

   PRESS:RETURN

   Please standby.. appears at the bottom of the screen, followed shortly by a **Writing File** message. ORACLE is now creating a new key file containing the sorted data from field #2. When it is finished, it returns to the **Search and Report** menu.

3) Now let's run a search and print the results again, to see what difference our sort has made. Go to **Design Reports** and get format #3 on the screen. Run a * search in the 'first name' field.

   When your printout is finished, compare it to the one you got before the field was sorted.

In our first printout the order of first names was:

Maria
Jose
Allan
Annie
Alicia
Albert
Betty
Ruby
Frankie
Al

They are listed according to record number. In the new printout, the order is (roughly):

Allan
Al
Albert
Alicia
Annie
Betty
Frankie
Jose
Maria
Ruby

We say 'roughly' because the order of the first four names on your list may not be exactly the same as those listed above. The list has been sorted alphabetically, but because the sort depth was 2, only the first two letters in each name were considered.

5) To get a thorough alphabetical listing of this field, our sort will have to be deeper. A sort depth of 3 will do the trick. Go back to the **Sort Fields** mode and re-sort the 'first name' field to a depth of 3. Now, run the same search. The order should be:

Al
Albert
Alicia
Allan
Annie
Betty
Frankie
Jose
Maria
Ruby

Our list of first names is now thoroughly alphabetized.

6) That kind of sort would be fine if all we were interested in was an alphabetized list. But what if we needed something a little more complex? For example, a list sorted by age and then alphabetized within each age. For that kind of problem, we need to look at subsorts.

First, go back to the **Sort Fields** mode, and sort the 'age' field to a depth of 2. Do not do any subsorts yet. Then go to format #3 again, and run a * search in the 'age' field to print your list. Here's what you'll get, again roughly:

Age	Name
13	Ruby Begonia
14	Jose California
14	Frankie Gezundtheit
14	Maria California
15	Al Emenopee
15	Betty Iguana
16	Alicia DeSoto
17	Annie Ablative
18	Allan Fliegel
19	Albert Galapagos

As you can see, ORACLE sorted everyone by age, all right, but volunteers of the same age appear in no particular order. This is why your list may not be exactly like the one here.

Unlike in our previous example, we can't sort the 'age' field any deeper to get a more refined sort. But we can do a subsort to list the volunteers alphabetically by first name whenever two or more are the same age.

7) Return to the **Sort Fields** mode again, and sort the 'age' field (#4) to a depth of 2. This time, when ORACLE asks if you want a subsort,

PRESS:y

for 'yes'. The bottom prompt changes to: **Enter field number to sort [1]**. We want to subsort first names, so

ENTER:2

ORACLE now asks how deep you want your subsort.

ENTER:4

ORACLE now asks: **More Subfields? [n].** You can specify as many as 9 subsort fields. We don't need any more, however, so

PRESS:RETURN

to default to 'no'. After sorting the field and writing the file, ORACLE returns to the **Search and Report** menu.

Now let's go back and do another * search in the 'age' field of format #3. Here's what you should get.

Age	Name
13	Ruby Begonia
14	Frankie Gezundtheit
14	Jose California
14	Maria California
15	Al Emenopee
15	Betty Iguana
16	Alicia DeSoto
17	Annie Ablative
18	Allan Fliegel
19	Albert Galapagos

The volunteers are arranged by age exactly as before, but now, wherever there is more than one volunteer of the same age, they are arranged alphabetically by first name as well.

ORACLE's capability to perform complex, multi-level sorts is very useful in large files where there may be many exact matches in a given field. A common example of this is the phone book, where hundreds of people with identical last names are subsorted by first name, initial, street name, and finally street number.

Lesson 14 - ORACLE Math [P]

ON SCREEN: Search and Report Menu

As you've seen, ORACLE allows you to look at your data in different ways by using different kinds of sorts and searches. But ORACLE's reporting power doesn't end there. ORACLE can also perform calculations with your numerical data and print out the results. ORACLE math takes place in the **Design Report** mode, using special field types in which the calculations are performed. In this lesson, we're going to look at ORACLE's four basic arithmetic functions -- addition, subtraction, multiplication, and division.

1) First, make sure your printer is on and set to print 55 lines per page, with the header: 'ORACLE Math'.

2) We want to calculate the average age of our volunteers, basing our report design on format #1. So to get started, get format #1 on your screen.

3) When the format appears, make the following changes:

   o On the top line, change the 101 field that contains the text 'COOKIES SOLD' to a 101 field that contains the text 'AGE'. Place this field at the right edge of the format.

   o On the fourth line, change the second field from data field #12 (cookies sold) to data field #4 (age). Again, place this field at the right edge of the format.

   o Delete the final line of the format.

   Now you have a format that will print a column of names and a column of ages.

4) The average age is the total of all the ages, divided by the number of volunteers. To start, then, we have to sum (add up) all the ages, and to do that we have to create a special report field. Move your cursor down a line and

   PRESS:<up arrow>

   to start a new field. The **Enter field #** prompt comes up. We're not particularly interested in printing out the sum of all the ages. The total age is merely an intermediate result on the way to our final goal -- average age. So we'll set up a non-printing variable field to hold the sum. Non-printing variable fields are always identified as field type 114.

ENTER:114

The **Enter variable field data** prompt appears. Into this field we will put the command for ORACLE to sum all the ages. All sum commands take the following general form: **'@sx#FP'**. Don't enter this; it's just a general way of stating a sum formula. Let's explain what these symbols mean:

- The **'@'** sign tells ORACLE that your variable field is to contain a calculation. The '@' always goes in the first space of a calculation field.
- The **'s'** tells ORACLE to perform a sum calculation (in other words, to add a bunch of numbers).
- The **'x'** tells ORACLE which **accumulator*** to use to hold your sum. An accumulator is a kind of electronic bin in your computer's memory. ORACLE dumps numbers into this bin and keeps adding them together. ORACLE can use up to 36 different accumulators, identified as 'a' to 'z' and '0' to '9'. Each different sum calculation within a single format must use a different accumulator.
- The **'#FP'** stands for **field position number*** (not to be confused with 'field number'). It tells ORACLE where to get the numbers for the calculation. The field position number represents, logically enough, the position of a particular field on the screen. You determine a field's position by counting your report fields on the screen from left to right and top to bottom. On our current screen, for example, the field containing the word 'NAME' is field position #1; the field containing the word 'AGE' is field position #2; and the field containing the first blank line is field position #4. The field containing the ages of our volunteers is field position #6 (even though it is record format field #4). This can be a bit confusing, so you have to think carefully when you're counting '#FP's.

Now that you know the form for sum calculations, let's enter our actual formula in our 114-type field.

TYPE:@

to tell ORACLE that this is a calculation field.

TYPE:s

to tell ORACLE that the calculation is a running sum.

TYPE:a

to tell ORACLE to hold its running total in accumulator 'a'. And finally,

TYPE:#6

to tell ORACLE to get the numbers for the sum from the sixth field on the screen (field position #6). Now

PRESS:RETURN

to close the field. Notice the new symbol that appears after the close field marker.

5) Now we need to create a field in which ORACLE will count the number of volunteers whose ages we're going to average.

To do this, you type '@count' into your field. As before, the '@' tells ORACLE that the field will contain a calculation. The 'count' tells ORACLE the type of calculation to perform -- to count the number of records in the report.

We could use another non-printing 114-type field. However, this time we'll ask ORACLE to print out the match count, so we'll use a 102-type field to contain it. And, of course, we'll create a field to label the result.

First move your cursor down two lines. Then

PRESS:<up arrow>
ENTER:102
TYPE:MATCHES:
PRESS:RETURN

Now move the cursor over a few spaces and

PRESS:<up arrow>
ENTER:102
TYPE:@count
PRESS:RETURN

to create the field that tells us how many of the records printed out match our search criteria.

6) We need one final calculation now - dividing the sum of the ages by the number of matches. This will be our 'average age' field. As with the previous field we will label it in the printout.

PRESS:<up arrow>
ENTER:102
TYPE:AVERAGE AGE:
PRESS:RETURN

Now, move the cursor over a few spaces and

PRESS:<up arrow>
ENTER:102

Now we have to enter the calculation. We will be dividing the final value in field position #9 (our sum field) by the final value in field position #11 (our count field), so we

TYPE:@#9/#11

where the '/' means 'divided by'.

PRESS:RETURN

to finish the field.

Any variable field in a report format can contain a calculation, but ORACLE allows only one calculation per variable field.

All calculations (except for sum and count) take the general form: **'@#FP operation VALUE'**. The following list explains how this works:

- o The '@' always tells ORACLE that the field is to contain a calculation.
- o The '**#FP**' stands for the field position number on the report screen.
- o The '**operation**' is the actual arithmetic procedure - addition, subtraction, multiplication, or division. The symbol for addition is '+'; for multiplication '*'; for subtraction '-'; and for division '/'.
- o The '**VALUE**' following the operation can be can a be field position number (#FP) or an actual number.

The following examples illustrate allowable calculations in ORACLE:

```
@#4+3     (field position #4 plus 3)
@#9*3.14 (field position #9 times 3.14)
@#12-#11 (field position #12 minus field position #11)
@#9/#11   (field position #9 divided by field position #11)
```

In ORACLE calculations, the '@' sign must always be followed by a field position number (#FP). Therefore, the following calculations are not legal:

```
@4-#13    (4 minus field position #13)
@12/#11   (12 divided by pield position #11)
```

But there's still a way to perfom these calculations. If you need to subtract a field number from actual number, or if you need to divide an actual number by a field position number, here's what you do. First, place the actual number into a variable field of its own (usually a non-printing **114-type** field), and then simply use that field position number as the first element in your calculation.

For example, you can subtract field position #13 from 4 by placing the number 4 into a **114-type** variable field (say, field position #10), and then using the calculation '@#10-#13'.

Similarly, you could divide 12 by field position #11 by placing the number 12 into a **114-type** variable field (say, field position #8), and then using the calculation '@#8/#12'.

Okay, with that out of the way, let's go back to our averaging example.

7) There are no more fields to be put on this page. To make the printout read neatly, go back into the last two lines and use your INST/DEL key to line up the right edges of your 'count' field and your 'division' field. That way the numbers printed in them will be lined up in your printout.

Your screen should now look like the one in Fig. 14.1

```
[NAME]                           [AGE]
[------------------------------------]
[ ]
[                     ]b          [  ]d
[ ]
[------------------------------------]

[@sa#6]

[MATCHES:]                [@count]

[AVERAGE AGE:]           [@#9/#11]
```

Fig. 14.1

If it does, and everything else is okay,

PRESS:<left arrow>

Check your format once more, just to be sure. Remember, if you don't have an equal number of open and close field markers, you'll get an **Error - 'open' & 'close' unmatched** message, and the cursor will flash at the problem area (or at the bottom of the screen if your last field is unclosed). If all is as it should be

PRESS:RETURN

for no more changes. We want to save this format, so

PRESS:RETURN
PRESS:4

to save it as format #4.

8) The cursor is now blinking in our first data field, and we can enter search data. Let's run a * search in

the 'name' field.

The first name to come up (because we sorted this field in the last lesson) is Al, aged 15. The 'match' field reads '1', and the 'average age' field reads '15'.

PRESS:n

to go to the next matching record. The name is Albert, aged 19. Total of both ages is now 34, there are 2 matches, and the average age is 17.

PRESS:c

to run the rest of the search continuously. At the end of the search, on your screen the total of all ages is 155, the number of matches is 10, and the average age is 15. (Don't worry, ORACLE can do decimal calculations; we just haven't got there yet.) On your printout the sum of the ages is missing because we put it in a non-printing 114 field.

9) Let's run another search using the same format, but different search data. This time, we'll look for the average age of all the volunteers whose names begin with any letter that comes after 'c'. Get format #4 back on the screen, and when the **Search Data** prompt appears, run a 'greater than 'c' search.

The first name to come up is Frankie, aged 14.

PRESS:c

to run the search continuously. There are only 4 matches here, with the average age coming out to 13.

By now you've noticed that ORACLE produced whole-number results where your pocket calculator would have given decimal results. That's because ORACLE will only give you decimal accuracy equivalent to the most precise decimal accuracy of the numbers used in your calculation. All our ages were expressed as whole numbers and our count was a whole number, so we got a whole-number result. If, on the other hand, we had used a mix of whole numbers and two decimal numbers in our calculation, we would have gotten a result accurate to two decimal places.

We can get a more accurate result for our average age by expressing the individual ages with greater precision. If we insert a field in our format which multiplies the 'age' field by 1.0, this new field will express ages accurate to one decimal place. If we now do our calculations referring to this new field, we will get one-decimal accuracy in our result.

Decimals must always be entered in ORACLE with a number before the decimal point (for example, 0.5 instead of .5).

10) Making the change is easy. Go back to format #4. Insert a non-printing 114-type field just above the sum field.

    ENTER: @#6*1.0

    This tells ORACLE to multiply the data from field position #6 by 1.0. This field must be inserted rather than placed at the end of the format because ORACLE reads the screen left to right, top to bottom. It can't refer forward to field position #14 to get data for a calculation that's taking place at field position #8. It can only reach back.

    Now, we have to change the reference in our sum field. Your sum field should now read '@sa#9', because we want it to sum the data from field position #9 now, not #6.

    Next, we have to change the references in our final field. By inserting a new field into our format, we've increased all the following field position numbers by one. So your final field should now read '@#10/#12'.

    There's just one small thing to do. Our sum field is currently 5 spaces long. In our previous format, that was quite sufficient to hold the 3-digit result of the operation. Now, however, we're going to get a longer number. On top of that, ORACLE always reserves an extra space in calculations for a possible plus or minus sign. So our sum field must now be 6 spaces long (three spaces for the whole number, one for the decimal point, one for the number after the decimal point, and one for ORACLE).

    If you ran a * search in the 'name' field without changing the length of the sum field, after a few records the sum field would fill up with question marks. This is ORACLE's way of telling you that there's not enough room in the field for the results of the calculation.

To make a calculation field longer, you add spaces to the right of the variable field data. So INSerT a space to the right of the '9' in the sum field.

You should now have a screen that looks like the one in Fig. 14.2.

```
[NAME]                           [AGE]
[----------------------------------]
[ ]
[                    ]b          [  ]d
[ ]
[----------------------------------]

[@#6*1.0]

[@sa#9]

[MATCHES:]              [@count]

[AVERAGE AGE:]           [@#10/#12]
```

Fig. 14.2

Run the same searches in the 'name' field that we did earlier. The * search should give you an average age of 15.5, and the >c search should give you an average age of 13.7.

Further Explorations

Even with our very simple database, there's practically no end to the different kinds of report formats and calculations you can produce. A good way to become more familiar with this is to experiment on your own: dream up some formats with calculations in them and see if you can make them work. Your aim should always be not just to get the right results, but to get a neat, readable printout as well.

For those of you who want a bit of a challenge, you should try to create a format that gives you all the following:

o The combined total number of boxes of cookies and candies sold on each route.

- The total number of boxes of cookies sold; the total number of boxes of candies sold; and the combined total number of boxes of cookies and candies sold.
- A gross sales figure, given that cookies sold at $1.25 a box, and candies at $1.75 a box.
- A net sales figure, given that the price included 7% sales tax (by dividing gross sales by 1.07).
- Total sales tax.

There are a lot of different ways to handle this -- try to figure one out on your own. (If you can't work it out, Fig. 14.3 shows one approach.) Below is a list of the field types we've encountered so far. Remember, each sum function must use a different accumulator. Good luck!

Field #	Type
0	variable field
1 - 99	data fields
101	top-of-page variable field
102	end-of-report variable field
114	non-printing variable field
128+ (1-99)	non-printing data fields

```
[Rt.#]   [Cook.]    [Cand.]       [Both]
[--------------------------------------]
[ ]
[  ]j     [    ]l    [    ]k    [@#8+#9]
[ ]
[--------------------------------------]
[TOTS:]   [@sa#8]   [@sb#9]    [@sc#10]
[ ]
[Cook. Gross:]         [@#14*1.25   ]
[Cand. Gross:]         [@#15*1.75   ]
[Total Gross:]         [@#19+#21    ]
[ ]
[NET SALES]            [@#23/1.07   ]
[ ]
[SALES TAX]            [@#23-#26    ]
```

Fig. 14.3

Net sales, by the way, were $1144.39 -- enough to buy some nice playground equipment.

Lesson 15 - Printing Labels [P]

ON SCREEN: Search and Report Menu

For complicated reports the **Design Reports** mode gives you the kind of full screen editing capability you need. Some kinds of reports, however, aren't that complicated. They're basically the same sort of thing over and over again. Perhaps the best example is mailing labels. ORACLE has a special procedure that bypasses the **Design Reports** mode entirely to let you print mailing labels quickly and easily.

1) We're going to print address labels from our file, so make sure your printer is on. **Label Printing** is 3 on the **Search and Report** menu.

   PRESS:3

   The prompt is **Use a saved format? [y]**. We don't have one, so

   PRESS:n

2) The screen changes. It is now titled **Define Label,** and the new prompt reads: **Number of lines/label (9 max)? [3]**. The default is 3, but we'll use 4 lines for each label -- a line for name, a line for street and apartment number, a line for city, and a line for code. So

   ENTER:4

   The next prompt asks: **Field number for line 1? [1]**. The default is 1, but we want to start with first name, so

   ENTER:2

   which is the number for our 'first name' field. A new prompt asks: **More fields in this line? [n]**. The default is 'no', but we want to put the 'last name' field on the same line, so

   PRESS:y

   The prompt disappears, and the cursor returns to the previous line, blinking to the right of our earlier entry. Again, the default is 1. Since that's the field we want,

PRESS:RETURN

The same series of prompts now begins for line 2. On this line we want data from our 'street' field (#5) and our 'apartment #' field (#6). So to complete the next line,

ENTER:5
PRESS:y
ENTER:6

The prompt line for line 3 now comes up. Notice that the label printing procedure allows a maximum of two fields per line.

On line 3 we want only the data from the 'city' field (#7), so

ENTER:7
PRESS:RETURN

For our last line, we want the data from our 'code' field (#8). So

ENTER:8
PRESS:RETURN

The next prompt asks: **Tab pos. of next label? [1].** ORACLE wants to know how many spaces to tab (from the beginning of the first label) before printing the next label. A common tab setting on sheets of gummed blank labels is 30, but any number is okay (just be sure you leave enough room for the longest line in the label to print completely).

ENTER:30

The next prompt asks: **How many labels per row? [1].** Most sheets of labels have 3 labels per row, so

ENTER:3

The next prompt asks: **How many blank lines? [1].** ORACLE want to know how many lines to skip between rows of labels.

ENTER:2

A new prompt asks **All entries O.K.? [y].** The default is 'y'. If you press 'n' (Don't do it!), the prompt returns to: **Use a saved format? [y],** and you'll have to start over (mailing-label formats cannot be edited). If everything is all right

PRESS:RETURN

to default to 'yes'. A new prompt appears, this time at the bottom of the page: **Break field no.? [0].** We'll talk about the uses of **breakfields*** in **ORACLE Design.** For now, just

PRESS:RETURN

to default to zero. ORACLE now asks if you want to save the format.

PRESS:RETURN

to default to 'yes'. Now you enter a format number. You can use the same format numbers you used in **Design a Report** without causing ORACLE any confusion. We'll use #1.

PRESS:1

3) The screen clears, and a single new prompt appears at the top: **Enter field # to search [1].** You can run your search in any field of your record. Your search field does not have to be one of the fields you want printed on the label. Run your search in the 'route #' field (#10).

ENTER:10

A new prompt asks: **Enter search text for field.** The cursor is flashing between a set of field markers, representing field #10.

TYPE:*
PRESS:<left arrow>

Your printer jumps to life and prints ten addresses, three per row, arranged in order of route #.

Before we leave **Label Printing** there's just more thing to point out. Notice that the printer has not left a bunch of empty space between the first and last names (even though the first name field is 20 spaces long in your record format).

ORACLE automatically suppressed all extra spaces after the first names. The same thing happened between the street address and the apartment number.

ORACLE will always suppress all unnecessary blank spaces in the first field of any line that contains two fields. This gives you a normal-looking address.

So much for label printing. A baby could do it.

Lesson 16 - Printing Forms [P]

ON SCREEN: Search and Report Menu

The last data management option we're going explore in this Tutorial is an ORACLE procedure that allows you to fill in pre-printed forms with data from your files.

Suppose you decided to enroll your volunteers in the North American Volunteer's Alliance League. According the membership requirements of NAVAL, you'd have to fill in a separate form for each applicant. Not such a big job for only ten volunteers, maybe, but what if you had a hundred, or a thousand, or more? The application looks like the one in Fig. 16.1.

NORTH AMERICAN VOLUNTEERS ALLIANCE LEAGUE

Application for Membership

Last Name: ____________________ Age:__
First Name:____________________

Address

Street:____________________
City: ____________________
Code: _____

Office use only: A__ B__ C__

Fig. 16.1

1) The ORACLE option for creating formats for printed forms is #4 on the **Search and Report** menu.

   PRESS:4

2) A prompt appears at the top of a blank screen: **Use a saved Format? [y].** We don't have a saved format for printer forms, so

   PRESS:n

At this point a ferocious-looking screen appears. But don't worry, it's trained.

The most obvious things on the screen are the three columns of numbers. These represent data field numbers. On a 40-columnn screen they run from 01 to 66, and on an 80-column screen they run from 01 to 99 (the maximum number of fields you can have in a single record). Also, each of these field numbers is actually sitting inside a field on the screen.

At the bottom of the screen a prompt reads: **Enter forms data.** At the top of the screen, ORACLE indicates data it requires. ORACLE wants to know:

- the **LINE** and **COL**umn position of each blank area on the form to be filled with a data field; and
- the **LEN**gth (or number of spaces available on the form) for that field.

In order to give ORACLE this information, you have to imagine a grid of lines (6 per inch) and columns (10 per inch) laid over the form. Something like FIG 16.2. Nevertheless, setting up your data will take a little trial-and-error before you get your format for each different form just right.

```
     0   0    1    1    2    2    3    3    4
     1...5....0....5....0....5....0....5....0

01   NORTH AMERICAN VOLUNTEERS ALLIANCE LEAGUE
02
03           Application for Membership
04   -----------------------------------------
05   Last Name: _______________      Age:__
06   First Name:_______________
07   -----------------------------------------
08   Address
09
10              Street:____________________
11              City:   ___________________
12              Code:   _____
13              ------------------------------
14
15   Office use only: A__  B__  C__
16   -----------------------------------------
```

Fig. 16.2

Let's see how the system works.

3) The cursor is blinking in the fourth space in the first field on the screen. The first blank on the NAVAL form is a space for last name.

The first thing to consider is the field number. The field numbers on the screen are actually default values. And, like all defaults, they can be changed. However, in this case the default value is fine (because our 'last name' field is field #1).

Now we have to tell ORACLE where to print this data field, and how long to make it. From the grid guide in Fig. 16.2, we can see that the space for last names is on row 5 and starts in column 12. It is 15 spaces long. This is the data we now want to enter.

First, we enter the LINE position. It can be any number from 1 to 99. So, with the cursor still in the fourth space in the first screen field,

TYPE:5

to enter the LINE position. Now

TYPE:<space>

to leave a blank space. (ORACLE needs a blank space between LINE, COLumn, and LENgth entries.) The COLumn position can be anything from 1 to 255. In our case, as we said, it is 12, so

TYPE:12

and then

TYPE:<space>

Finally, we'll put in the LENgth. This number <u>cannot</u> exceed the original length of your data field, although it can be shorter, as in this case.

TYPE:15

The first field on your screen should now look like Fig. 16.3

```
[01 5 12 15 ]
```

Fig. 16.3

This field is now complete, so

PRESS:RETURN

to jump to the next field. Here, we'll enter the data for the second blank line on the form -- age.

But first, a word about order. On the screen, the data for various printer lines can appear in any order whatsoever. You could even start with the bottom of the page and work backwards if you wanted to. However, whenever a single line <u>on your form</u> has more than one blank space to be filled, you must enter the coordinates of the spaces in order, from left to right.

For example, if in a single line of your form you had three blank spaces beginning at column 7, column 23, and column 56, respectively, the data for the space at column 7 must be entered <u>before</u> the data for the space at column 23, which must be entered before the data for the one at column 56. They would not, however, necessarily have to be entered in consecutive screen fields.

4) Now back to our second screen field. We have to enter the coordinates for the 'age' blank in our form. The 'age' field in our record format is field #4. The default data field in the second screen field, however, is 02, so we have to change it. Move the cursor back until it is blinking over the 2, and

   TYPE:4

   and then move over a space. Reading from our form, we see that the blank for age is on LINE 5, starts in COLumn 38, and has a LENgth of 2. So to enter the data

   TYPE:5<space>38<space>2

   The first two screen fields should now look like Fig. 16.4.

```
[01 5 12 15 ] [04 5 38 2  ]
```

Fig. 16.4

5) Now you're going to fill in the remaining blanks yourself. Enter the data for the remaining four blanks in the same order that they appear on the form.

   One tip before we go: ORACLE always reads an empty screen field as an 'end of format' signal. Therefore, you cannot skip a screen field when filling in your data. You must enter all your data in consecutive fields. So to position six data fields on your form, you have to use the first six fields on the screen.

   When you're done, the first six fields on your screen should look like Fig. 16.5.

```
[01 5 12 15 ] [04 5 38 2  ] [02 6 12 15 ]
[15 10 19 15] [07 11 19 15] [08 12 19 5]
```

Fig. 16.5

If everything in your format is okay, you can now

PRESS:<left arrow>

to finish off. ORACLE asks: **Any changes? [n].** If you have no changes to make,

PRESS:n

to answer 'no'. ORACLE asks you to **Please standby** for a moment, and then the **Save the Format? [y]** prompt appears. We want to save this format so

PRESS:RETURN

to default to 'yes'. The **Enter Format number** prompt appears.

TYPE:1

In a few seconds, your format disappears from the screen. In its place is the prompt: **Enter field# to search [1].** The field# can be for any data field in your record format, regardless of whether it appears in the printer format.

PRESS:RETURN

to default to #1. Data field #1 appears beneath an **Enter Search Text** prompt. To get our whole file let's run an * search.

PRESS:*
PRESS:<left arrow>

to run the search. The **Searching** prompt appears briefly, and your printer chugs to life, printing your data configured to fit our sample form. It should look something like Fig. 16.6

```
Ablative                   17
Annie



      1541 Diskdrive Lane
      Compucity, USA
      12367
```

Fig. 16.6

The continue search prompts are on the bottom of the screen. If you now run a continuous search, your printer will print the data from all of the records, arranged alphabetically by last name (because our #1 field is a key field), with the data from each record configured in exactly the same way.

<Any key> now takes you back to the **Search and Report** menu.

Lesson 17 - A Last Look At What You've Done

On Screen: Search and Report Menu

Finally, to finish off this Tutorial, let's look at our file list. Remember, **Show all files** is number 1 on the **Disk Utilities** menu.

When we last looked at this list, you'll remember, the only files on it were **TREATS.HDR, TREATS.REL, TREATS.KY01,** and **TREATS.KY10.** The list is considerably longer now. ORACLE has created a file for each of the formats you've saved, as well as for your two new key fields.

TREATS.FRC1 to **TREATS.FRC5** contain report formats 1 to 5. **TREATS.FRL1** contains your mailing label format. And **TREATS.FRF1** contains your printer form format.

Finally, **TREATS.KY02** and **TREATS.KY04** contain your new key files.

Well, there's no question about it. Congratulations are in order. You now know how to use all of ORACLE's essential data management functions. Not only that, but your community is well on the way to having its new playground.

The next step is up to you. You should now have the skills to create and use a data base on your own. Nevertheless, ORACLE has even more powerful functions available to help you analyze and manipulate your data. These are covered in detail in the following chapters.

Chapter 3 - System Management

Section 1 - Exiting to BASIC Page 118

Section 2 - Disk Utilities Page 119

Section 3 - Sequential Files Page 122

Section 4 - Modifying Record Formats Page 125

Section 5 - Working With Multiple Drives Page 127

Section 6 - Full Disks/Splitting Files Page 129

Section 7 - Support Programs Page 130

Section 8 - Interfacing with PaperClip Page 132

Section 9 - ORACLE Security Page 133

Chapter 3 - System Management

Introduction

The **ORACLE Tutorial** was concerned with data management: that is, with all the ORACLE procedures used to create, save, find, manipulate, and print database information. But sometimes you have to operate with files as a whole rather than with individual bits and pieces of data. You might, for instance, want to change the name of a file, transfer a file to another disk, or scratch a file from a disk. You might want to change the record format of a file or use data from a file in word-processed documents or letters.

The ORACLE procedures that allow you to do these things are known as system management procedures. These system management procedures are generally much simpler than most data management procedures. In this chapter we will describe ORACLE's system management procedures and explain when and how they should be applied. Briefly, they are:

Exiting to BASIC -- for leaving ORACLE and returning to your computer's normal operating system.

Disk Utilities -- for looking at the disk directory, scratching or renaming files, formatting or copying disks, and changing disk device numbers.

Sequential Files -- for creating files to be used by other programs, such as the PaperClip word processor.

Modifying Record Formats -- for restructuring databases without having to re-enter old data.

Working With Multiple Drives -- for reading from and writing to different disk drive numbers and devices.

Full Disks/Splitting Files -- for working with files too large for a single disk.

Support Programs -- for sorting very large files and inspecting data.

Interfacing with PaperClip -- for creating form letters, complex reports, and special documents incorporating data.

ORACLE Security -- for creating passwords to allow selected users different levels of access to data.

Section 1 - Exiting to BASIC

Some ORACLE system management procedures take place outside of the main ORACLE program. You can exit from ORACLE and return to your computer's normal operating system. **Exit to BASIC** is option 6 on the **Main Menu.** When you select this option, the cautionary prompt: **Are you sure? [n]** appears.

Pressing RETURN defaults to 'no', and ORACLE returns to the **Main Menu.**

Pressing 'y' exits from ORACLE and resets your computer to its original power-up mode. The **ready** prompt and flashing cursor appear on the screen.

To re-enter ORACLE's data management options, the program must be re-loaded.

Section 2 - Disk Utilities

ORACLE's **Disk Utilities** are contained in option 5 on the **Main Menu.** The **Disk Utilities** menu presents six choices: **Show all files, Change disk unit, Scratch files, Rename files, Format disk,** and **Copy a disk.**

1) **Show all files** is a reference function that displays a complete list of all program files, relative files, and sequential files contained on the current disk. At the end of the list, ORACLE displays a **BLOCKS FREE** message indicating the remaining unused disk capacity.

   All data filenames in the list are followed by ORACLE's standard indicators to designate their functions within the program.

   .HDR designates record format files (see Appendix for layout of **.HDR** files).

   .REL designates record data files (see Appendix for layout of **.REL** files).

   .KYxx designates sorted key files, where 'xx' is the number of the sorted field.

   .FRCx designates files containing report formats, where 'x' is the format number.

   .FRLx designates files containing mailing label formats, where 'x' is the format number.

   .FRFx designates files containing printed form formats, where 'x' is the format number.

   Program files and sequential files (see **Section** 3 of this chapter, pp.xx) appear without any special designators.

   If the **BLOCKS FREE** message does not appear below the file list, it means that the list is longer than a single screen-page. <Any key> will give you the next page of your list.

2) **Change Disk Unit.** In most systems the disk unit is device #8. ORACLE defaults to device #8, drive 0 (zero) when it is loaded. However, you may be working with a unit that has a different device number; you may be working with two units and have to designate a

different device number to access the second one; or you may want to specify which drive of a dual drive unit to work with. You do all of these things with the **Change Disk Unit** utility.

When you select **2** on the **Disk Utilities** menu, the prompt: **Disk Unit? [8]** appears. The default is 8. It can be reset to any value from 9 to 14. The usual device number for a second drive unit is 9.

After the device number is set a prompt appears asking: **Disk Drive? [0]**. The default is 0 (zero), but it can be reset to 1.

Attempting to set either the device number or the drive number to illegal values causes the **Disk Unit?** prompt to reappear and ask a for new entry.

After both device number and drive number have been set, ORACLE returns to the **Disk Utilities** menu.

3) **Scratch Files** allows you to delete whole files from your disk. When you select **3** on the menu, the prompt **Enter filename to DELETE** appears. Filenames must be entered exactly as they appear in the **Show all files** list, including designators. Each individual file must be scratched separately. When you hit RETURN to enter the filename, the **Standby** message appears briefly. When it disappears, your file is scratched and ORACLE returns to the **Disk Utilities** menu.

   CAUTION: There is no **Are you sure?** query in this scratch utility. Before scratching any file, double check to make sure that it is the correct one.

   Your master ORACLE program cannot be scratched using this function, not even by accident.

4) **Rename Files** allows you to change the name (but not the label) of files on your disk. When you select **4** on the menu, the prompt **Enter NEW, OLD filename** appears. New filenames must be entered exactly as you want them to appear on the file list, including designators. Old filenames must be entered exactly as they already appear. Type in the names separated by a comma, as the prompt requests. When you hit RETURN, the **Standby** message appears briefly. When it disappears, the file is renamed and ORACLE returns to the **Disk Utilities** menu.

All files pertaining to a single database must be renamed separately. If you rename a **.HDR** file without renaming the associated **.REL** file, or vice versa, you will not be able to select either the new name or the old in the **Set Filename** mode.

5) **Format Disk** gives you a quick and easy way to format new disks from within the ORACLE program. Formatting a disk gives it a name and a two-digit identification number by which your disk drive can recognize it. Disk names can be up to 16 characters long and must consist of unshifted letters only. ID numbers can be any two letters or numbers. Different disks may have the same name (which is useful if they deal with the same database), but each disk should have a unique ID number.

When you select #5 on your **Disk Utilities** menu, the prompt **Enter disk name, two char. ID** appears. You enter your name and ID number, separated by a comma. When you hit RETURN, the **Standby** message appears. When it disappears after a few moments, the disk is formatted and ORACLE returns to the **Disk Utilities** menu.

CAUTION: Formatting a used disk will erase all the files on that disk.

6) **Copy a Disk.** This utility is available to users of dual drives only. When you select **6** on the **Disk Utilities** menu, the prompt **Place new disk in drive 1 and master disk in drive 0. Press any key when done** appears. Insert the disk from which you want to copy in drive 0 (zero), and the disk to which you want to copy in drive 1. When you press <any key>, ORACLE formats the new disk to match the old one, and then copies all the files from the disk in drive 0 (zero) to the disk in drive 1. The old disk is unchanged in the process. While ORACLE is copying, it displays the **Standby** message. When this message disappears after a few moments, the copy is complete and ORACLE returns to the **Disk Utilities** menu.

To protect your data against accidental loss, it's a good idea to make copies of all your workdisks.

Section 3 - Sequential Files

ORACLE data is stored in **relative files***. Relative files are structured so that any individual piece of data can be found directly, even if it is buried deep in the middle of the file. Another kind of file is a **sequential file***, in which data is accessed by running through the whole file from beginning to end until the particular item is found.

Essentially, a sequential file is just a long, unstructured list of information. Files stored in this way are very efficient for certain system management procedures -- like modifying record formats, interfacing with word processor programs, splitting up files, or very sophisticated sorting. ORACLE gives you the capability to select and write data from your relative files into sequential files, whenever you need to.

Sequential files are created by setting up a report, and then 'printing' the report to your disk instead of to paper. Here's how it works.

1) Set the filename of the file whose data you want to write into a sequential file.

2) Using a normal printer setup, get a **List fields** printout for reference.

3) Go to **Printer Setup** in the **Search and Report** menu. In response to the first prompt, default the number of linefeeds to 0 (zero).

   The second prompt is **Printer Device number [4]**. Instead of entering your printer device number, enter the device number of the unit to which you want to 'print' your file. Usually this will be #8. This tells ORACLE to write your file onto the disk instead of printing it out.

   Once you've entered your disk drive device number, the third prompt asks: **Enter Filename?.** Here you name your sequential file with

   ENTER:<drive number:filename>

   On single drives, the <drive number:> is unnecessary. On dual drives, however, you must

   ENTER:0:<filename>

or

ENTER:1:<filename>

depending on the drive number to which you want to write.

The fourth prompt asks: **Ascii or Cbm Code?** Choose 'Cbm'.

The fifth prompt asks: **Do you want all spaces? [n].** Default to 'no'.

ORACLE takes you back to the **Search and Report** menu. Now you're set to write a sequential file.

5) Go to **Forms Printing** on the **Search and Report** menu. Answer 'no' to the **Use a saved format? [y]** prompt. The **Forms Printing** formatting screen appears.

You must fill in one screen field for each of the data fields in your record format that you want written into the sequential file. Remember, you are actually creating a report that will write to your disk. As in any report, you can use all or only some of the data fields from your relative file. You can have data fields appear more than once.

Each data field must be assigned a unique and consecutive LINE number. The numerical order of the LINE numbers will determine the order in which data fields are written into the sequential file, regardless of the order in which your data fields appear on the screen. For COLumn number, enter '1'; and for LENgth, enter the lengtn of the data field.

As with any other printed form format, you may save this format.

Once the format is complete, the search prompts come up. Use a search to produce the records you want in the order you want them. As you run the search, ORACLE **reads*** the specified data from your relative file and writes it into the new sequential file. The original relative file is not affected by this procedure.

When the search is completed, your sequential file is finished. Its name will now appear in your file list.

Note: If the sequential file you want to create is short enough (fewer than 22 fields), you can also produce it in the **Design a Report** mode. Here, you design a report format with the required fields, as in a normal report. However, each field must start at the extreme left side of the screen, and no more than one field may appear on a single line of the screen. Using the **Design a Report** mode gives you additional capabilties in creating sequential files:

- You can add fixed text fields to the sequential file by using 0-type fields in the format.
- You can run more complex searches (see **ORACLE Design, Section 6,** pp.xx)

Section 4 - Modifying Record Formats

Occasionally, you may need to change the record format of an already existing file. You may want to insert or delete fields, change field lengths, or change field attributes. To do this, you must create a new record format and then add the data from your original file. This would be very time-consuming if you had to add each record from the old file to the new one individually. However, it can be done very quickly using a sequential file.

1) To begin, create the record format for your new file, incorporating all the changes you need. You must give your new file a new name because ORACLE will not allow two files to have the same name. When the format is created, get a **List fields** printout for both old and new files.

2) Now, create a sequential file from your old data file.

   - To make sure your data fields are written to the new file in the right order, use the new field number for that data as the line number in the **Forms printing** screen you use to create the sequential file.

   - Wherever a brand new field appears in the new file (one that has no counterpart in the old file), simply skip the LINE number corresponding to that new field number in the record screen. This will create a blank field. However, if the new field is the last one in your new record format, you must enter something in the final LINE number. (Remember, ORACLE assumes the report is over when it encounters an unfilled field in the **Forms printing** screen.) In effect, you need a space-holder. Simply enter one of your previous fields (you can enter a field number more than once in a report) in the last LINE number of the report as the space-holder. It's a good idea to use one of your shorter fields as the space-holder so you can edit your new data field more easily later in the process.

   - Whenever a field from the old file does not appear at all in the new file, simply exclude its field number from the sequential file.

3) Now, re-set your filename to the new file, and go to

the **Add** mode. When the **File or Keyboard? [k]** prompt appears, select 'File'. A new prompt appears: **Enter Filename.** Type in the name of your sequential file and hit RETURN. (If you enter a filename that does not exist, ORACLE defaults back to the keyboard.)

ORACLE will automatically fill your new file with the data from the old, record by record, until your sequential file is exhausted.

CAUTION: ORACLE can only fill in the new file accurately if the LINE numbers in the sequential file format match exactly the field numbers in the new record format.

Section 5 - Working With Multiple Drives

Some ORACLE users will be using dual disk drives or more than one single or dual disk drive. Whenever there is more than one drive in your system, you must, on occasion, tell ORACLE exactly which drive you want to read from or write to. A few simple rules cover all situations.

1) When ORACLE is first loaded, it defaults to disk device #8, drive 0 (zero). This means that, unless you tell it otherwise, ORACLE will read from and write to disk device #8, drive 0 (zero).

2) Using the **Change disk unit** option in the **Disk Utilities** menu, you can set the device number and/or the drive number to new values. The device number can be set to values from 9 to 14. The drive number can be set to either 1 or 0 (zero).

3) If you are using a dual drive system, you may want to read a file from one disk and write it to another. In any of the procedures in which this might occur, you will be asked to enter a filename. If your drive number is defaulted to 0 (zero), and you want to write to drive 1,

   ENTER:1:<filename>

   in response to the **Enter filename** prompt. If your drive number is set to 1, and you want to write to drive 0 (zero),

   ENTER:0:<filename>

   in response to the prompt.

4) If you are using two single drives in tandem, each drive will have a different device number. If the drive from which you want to read is not device #8, reset the device number in the **Change disk unit** option in the **Disk Utilities** menu.

5) ORACLE always reads from and writes to the same drive unless told otherwise. To write to a drive other than the one from which you are reading, you must go to **Printer Setup.** When the prompt asks: **Printer device number [4]**, enter the number of the disk device to which you want to write. Then enter the name of the file you will be writing, select 'Cbm' code, and select or refuse the **Do you want all spaces? [n]** query.

6) If you are using two dual drives in tandem you must use a combination of disk utility commands and **Printer Setup** commands to specify the disk to which you want to write.

Section 6 - Full Disks/Splitting Files

Occasionally, you may try to write something to a disk for which there is no room.

If what you were trying to write was part of a relative file, ORACLE will write as many records as it can and display a **File too long** error message. Your relative file, though incomplete, is not harmed.

If what you were trying to write was anything but a relative file, ORACLE will display a **Disk Full** error message when it comes to the end of the disk. In this case the file is not valid. You must then exit ORACLE and validate your disk (see your disk drive manual for validation procedures).

If you are using a dual drive system, or if you are using two single drives in tandem, it is possible to split large data files and distribute them over two (or more) disks, using the following procedure.

1) Working from the **Create a File** mode, and using your old record format, save the format onto as many new disks as you need.

2) Create a sequential file, using a search that selects only part of the records from your file (for example, a >m search to produce the second half of the alphabet). Write this sequential file to your new disk.

3) Add data from the sequential file to the record format on the new disk (using the **File** option in the **Add records** mode). The sequential file can then be scratched. After the sequential files have been scratched, there will be plenty of room on the new disks for more records.

Using this procedure you could, for example, split an alphabetically sorted file onto two disks - one containing records from 'a' to 'm', and the other containing all records from 'n' to 'z'.

Section 7 - Support Programs

In addition to the main program, your ORACLE master disk contains five short support programs that allow you to inspect and sort ORACLE data. The programs are **REL-READ** (read relative files); **HDR-READ** (read header files); **KEY-READ** (read key files); **SEQ-READ** (read sequential files); and **SUPERSORT** (for sorting large files). Each of these programs must be loaded separately. See your computer manual for loading BASIC programs.

1) **REL-READ** allows you to read the unformatted contents of relative files. When you run the program, the prompt **Enter filename** appears. After you enter the name of the file to be read, the prompt **Record No. (from,to)?** appears. You enter the record number at which you want to start reading, followed by a comma, and then the record number at which you want to stop reading. The data of each of the records within that range are then displayed on the screen.

2) **HDR-READ** displays the statistics of creation and the complete field list of a given file. When you run the program, the prompt **Enter filename** appears. After you enter the name of the file, the screen displays the number of fields, the character length, and the number of pages in your record format. The screen also lists the field number, name, length, and type of each field in the record. 'Type' indicates the attributes and security level of each field.

    0 = unattributed
    1 = alpha
    2 = numeric
    4 = key
    16 = security level 1
    32 = security level 2
    48 = security level 3

   The number displayed for any field will be the sum of that field's individual type numbers. For example, an alpha, key, security level 2 field would have type 1 + 4 + 32 = 37.

3) **KEY-READ** allows you to read the sorted contents of key files. When you run the program, the prompt **Enter Filename,Field #** appears. After you enter the filename, followed by a comma, and then number of the key field you want displayed, the contents of the file

are displayed in sorted order. The record number of each entry is displayed in brackets.

4) **SEQ-READ** allows you to read the contents of sequential files. When you run the program, the prompt **Enter Filename,Field #** appears. After you enter the filename, the contents of your sequential file are displayed.

5) **SUPERSORT** is a special program allowing you to sort extremely large files that require too much computer memory for ORACLE to sort normally. The number of records that can be sorted depends on the depth of sort and the machine you're using (see **Appendix III** for table of maximum sort sizes).

If you attempt too large a sort, ORACLE displays the message **Out of sort memory - Creating unsorted file.** When this appears, ORACLE writes an unsorted key file. (This may take some time.) It is then possible to sort this key file outside of ORACLE, using **SUPERSORT.**

When you run this program, the prompt **Enter filename,field #?** appears. The field must already be a key field (that is, you must already have tried to run the sort within ORACLE). After the filename, followed by a comma, and the key field number are entered, a second prompt: **Enter subfield, length (0,0=quit)?** appears. Now you can enter the specifications for a subsort. For 'subfield' enter the field number of the field in which you want the subsort to occur, and for 'length' enter the desired depth of the subsort (the maximum depth is 15). There are 9 levels of subsort available. Whenever you respond to this prompt with '0,0', the sort will begin.

During the sort a screen message appears indicating how many passes through the fields the computer is making in order to do the sort.

When the sort is finished, you can reload ORACLE and use your sorted key file just as you would any normally sorted key file.

Section 8 - Interfacing with PaperClip

One of the most powerful options ORACLE gives you is the ability to interface data files with the PaperClip word processor. By combining ORACLE's ability to store and sort data with PaperClip's versatility in creating documents, designing reports, and formatting printouts, you have at your disposal the power and flexibility of programs costing thousands of dollars on some systems. For example, it's possible to automatically fill in addresses and salutations on form letters using selected and sorted data from your ORACLE files.

The procedure for interfacing the programs is quite simple.

1) Create a sequential file of the data you want PaperClip to use. Write this file to your ORACLE workdisk.

2) Load PaperClip into your machine, and load your form letter (or other document) containing the variable blocks to be filled in (see your PaperClip manual).

3) Now insert your ORACLE workdisk into the drive. When you enter any output command in PaperClip, the prompt **Fill variable blocks?** appears. Answer 'yes'. When the prompt **Variable filename?** appears, enter the name of your sequential file. In your printout, PaperClip will insert the appropriate data from each individual record into your form. One form letter will be printed for each record.

NOTE 1: Since the maximum possible length of a variable block in PaperClip is 80 characters, your sequential files should not contain data from fields more than 80 characters in length.

NOTE 2: For form letters and most documents, in creating your sequential file, you should suppress extra spaces by defaulting to 'no' in the **Do you want all spaces? [n]** prompt. If, however, you are using PaperClip to produce a report containing columns, do not suppress extra spaces.

Section 9 - ORACLE Security

ORACLE's security options allow multi-level controlled access to all or some of the data in a file. This feature is useful whenever certain fields within a file contain confidential information. In this case you may want to prevent some of the people using your data from seeing those fields. For example, the personnel department of a company might want to restrict access to salary information from general users of an employee file.

Creating a Secured Program

The Commodore 64 version of the ORACLE master disk contains two versions of the ORACLE program -- **Oracle 64.run** and **Oracle 64.sec.** The latter version contains the security option. In order to create a secured program and secured files, Commodore 64 users must first load **Oracle 64.sec.** On the 8032 master disk there is only one version of the ORACLE program, and it contains the security option.

After you load and run a security-option version of ORACLE, the prompt **Do you want security codes? [n]** appears. RETURN defaults to 'no'. Any response to this question produces the prompt **Do you have a dual disk drive? [n].** (On 8032 versions, this prompt does not appear.)

If you answered 'yes' to the first question and requested security codes, the next prompt reads **Enter Master code (4 char).** The master code (security level 3) gives the widest possible access to a secured file. Users of the master code can display data from any field and have total access to the **Modify Records** mode. To set the master code for your secured program, enter any four characters (except 'xxxx').

The next prompt reads **Enter level 1 (3 char).** The level 1 code gives holders access to all fields designated 0 (zero) or 1 in the **Create Record Format** mode. Level 1 users may view or print these fields, but they have no access to the **Modify Records** mode. To set the 'level 1' code, enter any three characters.

The next prompt reads **Enter level 2 (3 char).** The 'level 2' code gives holders access to all fields designated 0 (zero), 1, or 2 in the **Create Record Format** mode. Level 2 users may view or print these fields, but they have no access to the **Modify Records** mode. To set the 'level 2' code, enter any three characters (except those that were used for level 1).

Once the codes are set (or if you answered 'no' to **Do you want security codes? [n]**), a **Save Secured Program** message appears, with a flashing cursor beneath a ready signal. The secured program you have just created must be saved on a disk of its own. Take your program disk out of the drive, and insert a formatted fresh disk in drive 0 (zero).

ENTER:`save"0:<name>",8`

where <name> is the name under which you want to save your secured program. Once the program is saved, you can run it. As you can see, it runs normally, no longer asking for security information. Hereafter, whenever you want to use your secured program you must load it from this disk using the name you have just set.

Changing security codes is exactly the same as creating a secured program. To change your codes simply create a new secured program using different code characters.

Creating Secured Files

Using your secured program you can create a secured file. The creation process for a secured file is identical to that for an unsecured file, with a single step added -- you must specify a security level for each field you create. After you have chosen the attributes for a field, the prompt **Enter security level (1-3) or RETURN** appears.

- Pressing RETURN creates a 'level 0' (zero) field, accessible to all users for viewing and printing.
- Pressing '1' creates a 'level 1' field, accessible to holders of codes 1, 2, or 3 for viewing and printing.
- Pressing '2' creates a 'level 2' field, accessible to holders of codes 2 or 3 for viewing and printing.
- Pressing '3' creates a 'level 3' field, accessible only to holders of code 3.

Only holders of code 3 (the master code) have access to the **Modify Records** mode.

Using a Secured File

To use a secured file, the secured version of ORACLE must be loaded in your computer. Users must be assigned passwords according to the appropriate level of security.

In a secured version of the program, ORACLE will ask users for passwords at various points, regardless of whether or not the filename set is the name of a secured file.

CAUTION: Secured relative files can be read with the **REL-READ** program. To maintain security do not copy this program onto secured program disks.

Chapter 4 - ORACLE Design

Section 1 - Designing Your Own Applications Page 137

Section 2 - Creating Effective Formats Page 138

Section 3 - Direct Printer Commands Page 142

Section 4 - Multi-Page Printouts Page 143

Section 5 - Breakfields Page 145

Section 6 - Complex Searches Page 147

Section 7 - ORACLE Math Page 149

Section 1 - Designing Your Own Applications: Introduction

In designing ORACLE files and formats to work with your own database, the best way to work is backwards. Before you start to lay out your record format, think very carefully about the purpose of the file. Will the file be used mainly to display data on the screen, or will the data be printed out? Will you want calculations to be performed with the data, or will it be displayed and printed mainly as entered? What kind of information do you need to search for easily in your file? How do you want your records to be sorted? How large do you expect your file to be?

Make a complete list of all the fields you will need on your record, including their contents, their maximum possible length, and their names plus, any other out-of-field data you will want on the screen. Determine how many fields you'll need, and how discreet their information should be. Would it be better to have separate 'first name' and 'last name' fields, or will it be okay for the entire name to occupy a single field? How about an 'area code' field for phone numbers? Will you need a 'last update' field? Try to anticipate all your needs.

Once you know the contents of your file, arrange your fields on graph paper so you can count rows and columns. Assign fields to definite pages, and block out the appearance of each page in your record.

Determine the information your reports must contain, and how you want your printouts to look. Make a complete list of all the variable fields you will need, of all the calculations you want to make, and the data needed to make them. Then block out your report format screens on graph paper.

As a rule, it's best use a separate disk for each database you create, even if your file is relatively small. Disks are cheap, and you want to be sure you'll always have plenty of room on the disk for adding records. For very large databases determine in advance how you will divide them over several disks, taking care to leave yourself enough room on each disk for expansion of your files.

NOTE: If you are using a single drive, and you anticipate writing sequential files from your records, you must leave room for them on your disk. If a relative file and its associated format files use more than half (332 blocks) the space on a disk, you will not have enough disk capacity to write a sequential file which includes all your data.

Section 2 - Creating Effective Formats

ORACLE's power is substantially enhanced if you can create effective formats, both for records and reports. This means record and report formats that are easy to read and use on the screen, and report formats which generate easily readable printouts.

General Considerations

1) A spaciously laid-out format is much easier to use than one crammed full of fields and out-of-field data. Put only as much as you really need into each page of a format.

2) On screen, collect related data in blocks of fields that read logically. For example, in our Tutorial record format, we kept all the address data together in a block, and arranged much like an address on a piece of paper.

3) Wherever feasible, work in columns. Line up open and close field markers vertically to create neat columns of data. Do the same with field names and out-of-field comments.

4) Avoid making fields longer than the print width of your printer if you want to be able to design reports and printouts easily.

5) If a field will fit on a single line of your format, keep it on a single line. Do not wrap it around two lines.

6) Use reverse video, underlining, and graphic characters (see your computer manual) to help differentiate out-of-field comments from other screen information. Use screen lines and empty rows to separate blocks of data.

7) To make screens easier to edit, try to avoid having close field markers in the last screen column.

Record Formats

1) You have 9 pages and 99 fields available to you for each record format.

2) If you create a record format that is more than 13 blocks long (the equivalent of four 80-column screens packed full), ORACLE will write the format onto a disk, but will not be able to read it. To check the lengtn of your format, get a directory of your workdisk in BASIC (see your computer manual).

3) Every page of your format must contain at least one field. If you want to devote an entire page to out-of-field text, include a one-space blank field on the page as well.

4) In the **File Maintenance** mode, ORACLE accepts search criteria based only on the first page of a record. Design your record format so that all your most important search fields and your key fields are on the first page.

5) To help insure accuracy of data entry, make all fields that will contain only alphabetic characters alpha fields, and all fields that will contain only numbers numeric fields.

6) Create as few key fields as possible. Remember, all key files are scratched, resorted, and rewritten every time you leave the **Modify a Record** mode. In long files with many key fields this can be very time consuming. Special sorts are always available for any field in the **Search and Report** mode.

7) ORACLE takes as a field name the first 12 characters (on a 40-column screen) or the first 30 characters (on an 80-column screen) of any string that immediately precedes a field. The character string for a field name should not contain, or be followed by, more than 10 consecutive blank spaces. Field names need not occupy all the spaces available to them. (Names on a 40-column screen can be less than 12 characters. Names on an 80-column screen can be less than 30 characters.)

8) Out-of-field characters (including lines across the screen) which are not intended as field names, and which immediately follow a field, must be separated from the following field name by at least 10 spaces.

9) When designing a record format, be sure to create an individual field for each piece of data to be used in sorts or subsorts. For example, if you want to sort

records by last name, and subsort them by first name, you must have both a 'first name' field and a separate 'last name' field.

10) If you have a field containing amounts of money, don't enter a '$' sign inside the field. If you need the '$' sign in printouts, you can always include it as fixed text in a variable field of its own.

Report Formats

1) Use a printout of your field list to guide you when you are laying out a report format.

2) Column positions that line up on the screen will line up in printouts. To create neat columns of text in printouts, line up your open field markers on the screen (because text is left-justified within fields). To create neat columns of numbers in a printout, line up your close field markers on the screen (because numbers are right-justified within fields).

3) Blank lines or lines that contain only out-of-field characters are ignored by the printer. To force a blank line in printouts, use a 0-type variable field containing a single space. Place this field on a line by itself.

4) To force a blank line between your page header and any top-of-page fields, use a 101-type field (containing a single space) in the first screen line of your report format.

5) The line spacing of your report can be changed using the linefeed command in printer setup. To double-space a report simply add 1 to your normal linefeed, to triple-space add 2, and so on.

6) For reports involving calculations, get a printout of your report screen and number all the fields on paper for easy reference. This will make counting field positions much easier.

7) All field position numbers in calculation fields must refer to earlier fields on the screen. ORACLE cannot refer forward to later fields for data.

8) To save yourself time, paper, and aggravation, check the results of complex printouts on the screen before

you print them out.

9) **Formatting Extra-wide Printouts:** It's easiest to confine the printer width of reports to the width of your screen. However, users of 40-column screens will often need to format printouts for an 80-column page, and all users may occasionally need to format printouts that are even wider than 80 columns. It is possible within ORACLE to format wider-than-screen printouts using the 'semi-colon' command. A few rules govern the use of this command:

- Typing a ';' into a variable field on your report format screen forces your printer to print the following screen line before it starts a new print line.
- The ';' command can only be used in a variable field. If you want to use it between two data fields then you must create a single-space variable field to contain it.
- The ';' is a non-printing character; the printer simply ignores it. Therefore, whenever it is used, the column positions of all screen fields that follow it will print out one space to the left of their apparent position on the screen. You must accommodate this shift in your report design.
- The ';' command truncates the line on which it appears. Any characters following the ';' on the same line will not appear in your printout. Therefore, if you want all the characters on your line to print, place the ';' at the end of the line.
- Do not use the ';' in your variable fields as punctuation. It functions only as a printer command.

The trickiest part of designing any wide printout will be getting the columns to line up properly. Because single print lines can be spread out over several screen lines, and because the ';' is a non-printing character, it can be quite difficult to tell from the screen exactly how the printout will look. The easiest way to check your format is to do a sample printout using only a few records, and then go back to the screen to make corrections.

Section 3 - Direct Printer Commands

Printouts can also be designed using **direct printer commands***.

A **printer command*** is a two-character code that turns on or shuts off your printer's various functions. Printer commands can be used for many different functions -- to **force pages***, underline text, enhance text, change margins, or force carriage returns. Exactly what you can do using printer commands depends on the capabilities of your printer. See your printer manual for a complete list of your printer's capabilities and the commands that control them.

ORACLE allows you to imbed these two-character printer commands within your report formats. The form for imbedding printer commands is <up arrow><command>. Printer commands work only within variable fields. Here's how it works.

1) Create an appropriate variable field in the normal way by pressing <up arrow> and entering the variable field type.

2) Once you are in the variable field, press <up arrow>, followed immediately by the appropriate two-digit code. When used within a variable field, the <up arrow> prints on the screen as a normal up arrow.

3) At this point, you can either close the field or enter text, depending on your needs.

Printer commands such as those for underlining and print enhancement will stay in effect until you turn them off again, using another <up arrow> followed by the appropriate two-digit code. Printer commands such as those that force pages or carriage returns are single-action commands and don't have to be turned off.

You can use as many direct printer commands as you want, in as many variable fields as you want, and you can use them anywhere within a variable field. A single variable field may contain as many printer commands as will fit. Like the ';' command, however, direct printer commands are non-printing characters. Therefore, each one you use pushes the column position of characters on your screen three spaces to the right of their column position on your printout. You must accommodate this shift in your report design.

Section 4 - Multi-Page Printouts

ORACLE allows you to create multi-page reports. ORACLE will properly number the pages and can print page totals and/or other remarks on each page. Multi-page reports are set up using variable fields. Variable fields can contain fixed text, calculations, direct printer commands, blank spaces, or the '?' as the first character in order to allow data entry during printing.

1) Field Types: There are four basic variable field types for formatting multi-page printing. As well, there are ten other variable fields available to clear accumulators at the end of each page.

   - **100-type** variable fields print only at the beginning of a report. They are most commonly used to print report titles and headings.
   - **101-type** variable fields print only at the beginning of each page. They are most commonly used for page headings and column labels.
   - **102-type** variable fields print only at the end of a report. They are most commonly used for end-of-report totals and summaries.
   - **103-type** variable fields print only at the end of each page. They are most commonly used for page totals and running sub-totals.

   NOTE: 101-type and **103-type** fields will only print if a page header or page numbering has been set during the **Printer setup** routine.

2) Page Accumulators: Normally, accumulators will keep summing values until the end of a report. For producing page totals, however, it is necessary to 'clear' accumulators after printing. This resets the accumulators to zero for the next page.

   Field types **104** through **113** are used for this purpose. They function exactly like **103-type** fields in that they print only at the end of each report page. However, when they print, they also clear (or reset to zero) specific accumulators associated with them. The accumulators associated with **104-type** to **113-type** fields are accumulators 'a' through 'j' respectively. In other words **104-type** fields clear accumulator 'a', **105-type** fields clear accumulator 'b', and **113-type** fields clear accumulator 'j'.

Normally, you would use these field types to print their associated accumulators. And in your report format, for example, you would enter [@sc] in a **106-type** field to both print and clear the sum of accumulator 'c'. However, when they print, these special field types always clear their associated accumulators. This means, for example, that even if you have a blank space in a **110-type** field, accumulator 'g' will clear at the end of the page.

Like **103-type** fields, field types **104** to **113** will only print if a page header or number has been set during the **Printer setup** routine.

3) Numbering pages: Pages in multi-page printouts will be automatically numbered if you include the '#' symbol in your page header during **Printer setup.** The page number will appear on the printout at the same column position that '#' was typed on the screen. Your printer should always be set with a page header if you expect to produce properly paginated multi-page reports.

4) Setting Page Length: If you've created a report format in which the data from each record occupies more than one printed line, ORACLE always insures that your pages contain only complete records, rather than splitting a record over two pages. Nevertheless, it's still a good idea to determine the best printer page length for these kinds of reports. To do this, decide the maximum number of records you want per page, and then count the total number of lines you need to print such a page, including page headers, footers, and blank lines. Then enter this number as your 'lines per page' number during the **Printer setup** routine.

Section 5 - Breakfields

A **breakfield*** is a special variable field that prints only at specified points in a report. In effect, it 'breaks' into the body of a report so that fixed text can be inserted, printer commands used, calculations performed, or accumulators cleared.

The breakfield 'breaks' the report whenever a specified piece of data changes. You might use a breakfield, for example, in a long report listing names alphabetically. If you wanted to start a new page for each letter of the alphabet, you could set your breakfield to force a page when the 'A's changed to 'B's, 'B's changed to 'C's, and so on.

You might also use a breakfield in a sales record file. If you wanted to print monthly sales totals, you could set your breakfield to print the sum of the sales whenever the month in your 'date' field changed.

Breakfields are designated as **115-type** fields. They can contain any kind of variable field information: text, printer commands, calculations, blank spaces, or the '?' symbol. Here's how they work.

1) To create breakfields enter '115' as the field number when the report format prompts: **Enter field # for position.** As for any variable field, ORACLE then prompts: **Enter variable field data & RETURN.**

2) After your variable data has been entered, the prompt **Break field no.? [1]** appears. The default is 1. Enter the data field number in your record format to which you want the breakfield to refer. Usually, this will be a key field (or a subsort field within a key field).

3) Now, the prompt: **No. of chars. (10 max)? [3]** appears. The default is 3. Enter the number of characters you want ORACLE to look at in determining the **breakpoint***. For example, if you wanted to force pages at every new letter of the alphabet, you would enter '1', and ORACLE would break each time the first letter in the field changed. If you wanted to break each time the month in a YYMMDD 'date' field changed, you would enter '4', and ORACLE would break each time the fourth character in the field changed.

ORACLE now returns to the normal **Enter format** prompt.

You can use as many breakfields as you want in your format, to contain all the text and calculations that you may need. However, the breakpoint can only be set once, on the very first use of a **115-type** field.

Breakfields can be used to clear accumulators 'a' through 'm'. If you use a breakfield to print a sum function using any of these accumulators ('a' through 'm'), that accumulator will clear after printing and will be set to zero for the next break. Breakfields clear only the accumulators they print, and breakfields will not clear accumulators 'n' through 'z' or '0' through '9'.

The column position of a breakfield in a printout will reflect its column position on the screen. However, no matter what screen line breakfields are entered on, they will print out on consecutives lines at the breakpoint in your report. So it's easy to append breakfields to the end of a report format -- they can even follow end-of-report fields on the screen.

ORACLE also provides a breakfield option in the **Label printing** mode for mailing labels. In this case ORACLE will automatically print a full line of *'s across the page, separating one group of addressing from the next.

Section 6 - Complex Searches

In the Tutorial we looked only at the basic types of searches you can run in ORACLE -- string searches, wild card searches, unequal match searches, and any match searches. We ran searches using only a single search condition.

It is possible in ORACLE, however, to run complex searches using more than one set of search data. There are two ways of doing this -- by using **logical-and*** searches, or by using **subfiles***.

Logical-and Searches

In the **View Records**, **Modify a file**, and **Design Reports** modes it is possible to run logical-and searches. Logical-and searches use two sets of search data. After entering the first search condition, instead of pressing <left arrow> press the '+' key. A '+' sign will appear on the top line of your screen (following the unequal match sign if your first search data specified an unequal match). You can now enter a second search condition. Press <left arrow> when you are done. ORACLE will now search for all the records in your file that conform to both conditions.

Logical-and searches always use data from two different fields. In accessing the data, ORACLE always refers to the topmost field on the screen. Therefore, if the topmost field of a logical-and search is a key field, ORACLE will display the data in the sort order of that field.

Logical-and searches cannot be run in the **Labels printing** or **Forms printing** modes.

Subfile Searches

It is possible to run searches with an unlimited number of search conditions using the subfile method. Subfile searches are more time-consuming than simpler searches, but they are also much more powerful.

A subfile of a given file is a separate file that uses the same record format as the original file, but which contains only some of its records. Here's how it works.

1) To create a subfile, first create a new file using the record format of your original. Save this subfile under a unique name.

2) Create a sequential file using the data from the original file. If your file is less than 22 lines in length, you can use the **Design a report** mode to do this; if it is longer, you must use the **Forms printing** mode.

3) Run any legal search to create your sequential file. (If are using the **Design a report** mode, this includes a logical-and search.)

4) Add the sequential file (using the 'file' option) to your subfile format. This creates a file that is formatted exactly as the original, but which contains only the records matching your original search criteria.

This four-step process can be repeated as many times as necessary to produce your desired result. By creating a series of files, subfiles, sub-subfiles, and so on, all using different search criteria, it is possible to run a search using as many different search conditions as you want. Using this method to analyze the data in a questionnaire, for example, you could produce a list of all the left-handed men, 30 to 39 years old, who lived in Tulsa, were baseball fans, ate peanut butter sandwiches more than 3 times a week, and owned digital watches (assuming, of course, you had asked all those questions in your survey).

Section 7 - ORACLE Math

ORACLE provides four arithmetical functions for performing calculations with numerical data -- addition, subtraction, multiplication, and division. ORACLE also provides 36 accumulators for summing values contained in report fields. As well, ORACLE provides a 'count' function which counts the number of records in a report which match the search criteria. These calculation options are available in the **Design a report** mode.

Arithmetic Calculations

1) All calculations must occur within variable fields.

2) Only one mathematical operation is allowed per field.

3) All variable fields containing calculations must have the '@' symbol in the first space of the field.

4) All field position numbers in calculation fields must refer to earlier fields on the screen. ORACLE cannot refer forward to later fields for data.

5) All decimal values must be preceded by a whole number value ('0.5' is a legal number; '.5' is not)

6) Calculation results are accurate to the degree of decimal precision of the values used to perform the calculation. There are two ways to increase decimal precision. You can add 0.0, 0.00, 0.000, etc., to one of the input values; or you can multiply one of the input values by 1.0, 1.00, 1.000, etc. Either of these methods will work, but adding is generally a faster operation for the computer.

7) Calculation fields must be long enough to display the result, including the integer portion of the number, the decimal point (if any), the decimal portion of the number (if any), and a possible plus or minus sign. The number 1.234, for example, requires a field at least 6 spaces in length. When a field is not long enough to hold the entire number, it will display **?????** when reporting. In order to increase the length of a variable calculation field, simply add blank spaces after the formula. Note, however, that the maximum length of a calculation field cannot exceed 17 spaces.

8) There are four allowable formats for calculation fields within ORACLE:

```
[@#FPx op #FPy]
[@#FPy op #FPx]
[@#FPx op #FPx]
[@#FPx op val]
```

where '#FPx' and '#FPy' are the field position numbers of fields 'x' and 'y'; 'op' is a mathematical operation; and 'val' is an unsigned numerical value.

9) For calculations requiring a fixed numerical value preceding the operator (as in subtracting from a fixed value or dividing a fixed value), first place the fixed value in a non-printing **114-type** field, and then use this field in the formula.

10) If for any reason, a calculation field contains an error, the field will display the calculation formula instead of the numerical result while reporting, and a **Syntax error** message will be displayed.

Accumulators

ORACLE provides 36 accumulators for summing values during a report. These accumulators are designated 'a' through 'z', followed by '0' through '9'.

1) The format for producing a sum in a given accumulator is '[@sx#FP]', where 's' tells ORACLE to perform a sum calculation, 'x' represents the accumulator used to hold the sum, and '#FP' is the field position number from which the values to sum are taken.

2) Each sum must have a unique accumulator.

3) Normally, accumulators will keep summing values until the end of a report. For producing page totals, however, it is necessary to 'clear' accumulators after printing. This resets the accumulators to zero for the next page. ORACLE uses accumulators 'a' through 'j' for this purpose.

4) End-of-page field types **104** through **113** clear accumulators 'a' through 'j' respectively. In other words, **104-type** fields clear accumulator 'a', **105-type** fields clear accumulator 'b', and **113-type** fields clear accumulator 'j'. When these fields print, they

automatically clear the specific accumulators associated with them.

5) Breakfields can be used to clear accumulators 'a' through 'm'. If you use a breakfield to print a sum function using any of these accumulators ('a' through 'm'), that accumulator will clear after printing and will be set to zero for the next break. Breakfields clear only the accumulators they print, and breakfields will not clear accumulators 'n' through 'z' or '0' through '9'.

Match Count

ORACLE's match count function produces a count of the number of records matching the search criteria in a report. The format for this function in a variable field is '[@count]'. (Although '[@c]' is sufficient, providing the result will fit into such a small field.)

Part II - The ORACLE Reference Guide

Section 1 - Reference Guide to ORACLE Functions Page 153

Section 2 - Appendices: Page 200
- I - Inserting Your ROM Chip Page 201
- II - ORACLE Special Keys Page 202
- III - Sort Chart Page 203
- IV - Security Access Chart Page 204
- V - ORACLE Field Types Page 205
- VI - ERROR Messages Page 207
- VII - .HDR and .REL File Layout Page 209
- VIII - Changing Default Values Page 211

Section 3 - Glossary Page 212

Section 4 - Index Page 218

Section 1 - Reference Guide to ORACLE Functions

The following guide is a complete reference list of the keystroke sequence for each ORACLE function.

Functions are listed alphabetically by name. Each reference also contains a page number in the manual indicating where the function is described in detail. The first part of the function name is the mode, outcome, or screen display with which the function is concerned. The second part of the name is the operation. For example, **Set Filename** is listed as **Filename: Set.**

Each function is broken down into the individual steps required to perform it. Each step is numbered.

In many of the steps, you will see the term 'go to', followed by another step number. When you encounter a 'go to', simply move to the step number referred to and continue from there.

If no 'go to' is indicated, then simply go to the next step.

In some of the steps, you will see the term 'perform function', followed by the name and number of another function in the reference section. This means perform the function indicated and then return to where you were.

ORACLE Functions

1. Breakpoint: Specify Page 155
2. Disk: Copy Page 155
3. Disk: Format Page 156
4. Disk Drive: Change Drive Number Page 156
5. Disk Unit: Change Device Number Page 157
6. Escape from Function Page 157
7. Exit to BASIC Page 157
8. Field List: Print Page 158
9. Fields: List Page 158
10. Fields: Sort Page 158
11. Filename: Set Page 160
12. Files: Change Format Page 160
13. Files: Create Page 161
14. Files: Rename Page 163
15. Files: Scratch Page 164
16. Files: Show All Page 164
17. Forms: Format and Print Page 165
18. Labels: Format and Print Page 168
19. Menus: Display Page 174
20. Passwords: Change Page 175
21. Passwords: Enter Page 175
22. Printer: Setup Page 177
23. Printer: Shut off Page 178
24. Programs: Backup Page 179
25. Programs: Load Page 179
26. Records: Add from Keyboard Page 180
27. Records: Add from Sequential File Page 181
28. Records: Change Page 182
29. Records: Delete Page 183
30. Records: Print Page 184
31. Records: View Page 186
32. Reports: Format and Print Page 187
33. Screen: Print Page 191
34. Searches: Interrupt Page 191
35. Searches: Run Page 192
36. Secured File: Create Page 196
37. Secured Program: Create Page 197
38. Sequential File: Create Page 198

1. Breakpoint: Specify pp.145-6

A breakpoint can only be specified from within the **Reports: Format** or **Labels: Format and Print** function. (For **Label** breakpoints, see **18. Labels: Format and Print.**) From within **Reports: Format**, use the folowing steps.

1.1 Create a 115-type field.

TYPE:<up arrow>
ENTER:115
ENTER:<variable field data>

1.2 Prompt: **Break field no.? [1]**

If the breakfield is to refer to data field #1,

PRESS:RETURN

If the breakfield is to refer to some other data field,

ENTER:<data field number>

1.3 Prompt: **No. of chars. (10 max)? [3]**

If the breakfield is to refer to the first three characters of the data field,

PRESS:RETURN

If the breakfield is to refer to a different number of characters (1 to 10 allowed),

ENTER:<number of characters>

2. Disk: Copy pp.14-15

The disk copy utility can only be used with dual drives.

2.1 On Screen: **Main Menu**

PRESS:5

2.2 On Screen: **Disk Utilities Menu**

PRESS:6

2.3 Prompt: **Place new disk in drive 1 and master disk in drive 0. Press any key when done.**

Place disks in drives and

PRESS:<any key>

3. Disk: Format pp.21-2

3.1 On Screen: **Main Menu**

PRESS:5

3.2 On Screen: **Disk Utilities Menu**

PRESS:5

3.3 Prompt: **Enter disk name, two char. ID?**

ENTER:<disk name>,<ID>

4. Disk Drive: Change Drive Number pp.119-20

4.1 On Screen: **Main Menu**

PRESS:5

4.2 On Screen: **Disk Utilities Menu**

PRESS:2

4.3 Prompt: **Disk Unit?**

PRESS:RETURN

4.4 Prompt: **Disk Drive?**

ENTER:<new drive number - 0 or 1>

5. **Disk Unit: Change Device Number** pp.119-20

5.1 On Screen: **Main Menu**

PRESS:5

5.2 On Screen: **Disk Utilities Menu**

PRESS:2

5.3 Prompt: **Disk Unit?**

ENTER:<new device number - 8 to 14>

5.4 Prompt: **Disk Drive?**

ENTER:<drive number on new unit - 0 or 1>

6. **Escape from Function** p.19-20

The **Escape** function returns you to the **Main Menu** from any other function, as long as ORACLE is loaded and there is a flashing cursor on the screen.

6.1 On the Commodore 64,

PRESS:f8

6.2 On all other models,

PRESS:(shift)RUN/STOP

7. **Exit to BASIC** p.118

7.1 On Screen: **Main Menu**

PRESS:6

7.2 Prompt: **Are you sure? [n]**

If no,

PRESS:RETURN

to return to **Main Menu.**

If yes,

PRESS:y

to exit program.

8. Field List: Print p.73

8.1 Perform function **22. Printer: Setup**

8.2 Perform function **9. Fields: List**

9. Fields: List p.41

9.1 On Screen: **Main Menu**

PRESS:4

9.2 On Screen: **Search and Report menu**

PRESS:6

9.3 Prompt: **Strike any key to continue.**

To return to **Search and Report** menu

PRESS:<any key>

10. Fields: Sort pp.88-92

10.1 On Screen: **Main Menu**

PRESS:4

10.2 On Screen: **Search and Report menu**

PRESS:7

10.3 Prompt: **Enter field number to sort - [1]**

If you want to sort field #1,

PRESS:RETURN

If you want to sort any other field

ENTER:<field number>

10.4 Prompt: **Enter sort depth (15 max.) - [5]**

If you want a sort depth of 5,

PRESS:RETURN

If you want another depth of sort, from 1 to 15 characters,

ENTER:<sort depth>

10.5 Prompt: **Subsort req'd? [n]**

If no subsort is required

PRESS:RETURN

The field will now be sorted. If a subsort is required,

PRESS:y

10.6 Prompt: **Enter field number to sort - [1]**

If the subsort field is field #1,

PRESS:RETURN

If the subsort field is not field #1,

ENTER:<subsort field number>

10.7 Prompt: **Enter sort depth (15 max.) - [5]**

If you want a sort depth of 5,

PRESS:RETURN

If you want another depth of sort, from 1 to 15 characters,

ENTER:<sort depth>

10.8 Prompt: **More subfields? [n]**

If no more subfields are required,

PRESS:RETURN

Your field will now be sorted and subsorted. If more subsorts are required,

PRESS:y

and go to 10.6.

11. Filename: Set pp.39-40

11.1 On Screen: **Main Menu**

PRESS:1

11.2 On Screen: List of relative files.
Prompt: **Enter Filename?**

ENTER:<portion of relative file name preceding **.REL**>

12. Files: Change Format pp.125-6

12.1 Perform function **13. Files: Create.**

Create a new file format based on the old format, incorporating any changes you want to make.

12.2 Perform function **38. Sequential File: Create.**

Create a sequential file of all the fields in the old data file that are to appear in the new file, in the same order that they are to appear in the new file. The sequential file format must contain the same number of fields as the new file. Wherever the new file contains a field that does not appear in the old file, it must be inserted in the sequence of fields in the sequential file.

12.3 Perform function **27. Records: Add from Sequential File.**

Add data from sequential file to new data file.

13. Files: Create pp.24-37

13.1 On Screen: **Main Menu**

If the format of the file to be created is to be based on the format of an existing file, the existing filename must be set. To set existing filename perform function **11. Filename: Set.** When the filename is set, go to 13.2.

If the format of the file to be created will not be based on the format of an existing file, go to 13.2.

13.2 On Screen: **Main Menu**

PRESS:2

13.3 Prompt: **Enter 1 for old or 2 for new format.**

If the format of the file to be created is to be based on the format of an existing file,

PRESS:1

Go to 13.6

If not,

PRESS:2

Go to 13.4

13.4 Prompt: **Enter <up arrow> to start field and <left arrow> when done**

Only out-of-field characters can be typed onto the screen if this prompt is displayed.

To create a field, move cursor to the screen position for the start of the field, and

PRESS:<up arrow>

Prompt: **Enter length: xxxx max. and return**

ENTER:<length of field>

Prompt: **Alpha, Numeric or Key field?**

If this field is to have attributes,

ENTER:<attributes - a, n, k, ak, or nk>

If this field is to be unattributed,

PRESS:RETURN

13.5 Prompt: **Enter <up arrow> to start field and <left arrow> when done**

If you want another field, go to 13.4.

If you do not want another field, add any remaining out-of-field characters to format if desired, and then

PRESS:<left arrow>

13.6 Prompt: **Record Length=XXXX all fields o.k.? [y]**

The number indicates the total length of all the fields on this page.

If there are any changes to be made,

PRESS:n

and go to 13.4.

If there are no changes,

PRESS:RETURN

13.7 Prompt: **Would you like another page? [n]**

If you do not want another page

PRESS:RETURN

and go to 13.9.

If you would like another page

PRESS:y

13.8 On Screen: Next Page
Prompt: **Enter <up arrow) to start field and <left arrow> when done**

To create page, go to 13.4.

To cancel page

PRESS:<left arrow>

Prompt: **Cancel this page? [y]**

To affirm page cancelled

PRESS:RETURN

and go back to 13.7.

To escape from cancellation command

PRESS:n

Prompt: **Error: No fields specified**

PRESS:<up arrow>

and go to 13.8.

13.9 On Screen: **Statistics of Creation**
Prompt: **Enter Filename**

Filenames must be ten characters or less, and must not be punctuated by a period.

ENTER:<filename>

14. Files: Rename pp.120-1

14.1 On Screen: **Main Menu**

PRESS:5

14.2 On Screen: **Disk Utilities menu**

PRESS:4

14.3 Prompt: **Enter NEW, OLD filename**

ENTER:<new filename>,<old filename>

Both old and new filenames must include their designators (such as '.REL' or '.HDR').

15. Files: Scratch p.120

15.1 On Screen: **Main Menu**

PRESS:5

15.2 On Screen: **Disk Utilities** menu

PRESS:3

15.3 Prompt: **Enter filename to DELETE**

ENTER:<name of file to be deleted>

Filenames must include their designators (such as '.REL' or '.HDR').

16. Files: Show All p.119

16.1 On Screen: **Main Menu**

PRESS:5

16.2 On Screen: **Disk Utilities** menu

PRESS:1

16.3 Prompt: **Strike any key to continue.**

If there is no **BLOCKS FREE** message on the screen, your file list occupies more than one page. To see the next page

PRESS:<any key>

and go to 16.3.

If there is a **BLOCKS FREE** message on the screen,

PRESS:<any key>

to return to the **Disk Utilities** menu.

17. Forms: Format and Print pp.108-14

17.1 On Screen: **Main Menu**

If forms are to be printed perform function **22. Printer: Setup.** Then go to 17.2.

If forms are to be formatted, but not printed

PRESS:4

17.2 On Screen: **Search and Report menu**

PRESS:4

17.3 Prompt: **Use a saved Format? [y]**

If no

PRESS:n

and go to 17.5.

If yes

PRESS:RETURN

17.4 Prompt: **Enter format number -**

TYPE:<number of saved format to be used>

On Screen: Saved Format
Prompt: **Any changes? [n]**

If no changes desired,

PRESS:RETURN

and go to 17.15.

If changes are required,

PRESS:y

17.5 On Screen: **Form format screen, with or without saved format.**

The procedures for creating a new format or modifying an old format are the same. Data must always be entered in consecutive fields on the screen. ORACLE interprets an unfilled screen field to mean the end of the format.

Prompt: **Enter forms data (<left arrow> = end)**

The cursor is blinking in the first screen field. If creating a new format, or if existing data in this field is to be changed, go to 17.7.

If no data is to be changed in this field go to 17.6.

17.6 To move cursor to the next field,

PRESS:RETURN

If forms data is to be added or changed in this field go to 17.7.

If no data is to be changed in this field go to 17.6.

17.7 Prompt: **Enter forms data (<left arrow> = end)**

If data field number need not be changed, go to 17.8.

If data field number must be changed, move cursor back to first space in screen field and

TYPE:<new data field number>

17.8 Prompt: **Enter forms data (<left arrow> = end)**

You may or may not leave a blank space between the field number and the line number.

If existing line number need not be changed, go to 17.9.

If a line number must be added, or if existing line number needs to be changed, move your cursor to the third or fourth field space and

TYPE:<one- or two-digit line # - 1-99 allowed><space>

17.9 Prompt: **Enter forms data (<left arrow> = end)**

If a previous screen field contains the same line number as the field into which you are currently entering data, then the column number entered here must be greater than the column number in the previous field.

If existing column number need not be changed, go to 17.10.

If a column number must be added, or if existing column number must be changed,

TYPE:<column # - 1-255 allowed><space>

17.10 Prompt: **Enter forms data (<left arrow> = end)**

If existing field length number need not be changed, go to 17.11.

If a field length number must be added, or if an existing field length number needs to be changed

TYPE:<field length # - equal to or less than data field length>

17.11 Prompt: **Enter forms data (<left arrow> = end)**

If there are other screen fields to be added or modified, go to 17.6.

If there are no other screen fields to be added or modified, go to 17.12.

17.12 Prompt: **Enter forms data (<left arrow> = end)**

PRESS:<left arrow>

Prompt: **Any changes? [n]**

If changes are required,

PRESS:y

and go to 17.5.

If no changes are required,

PRESS:RETURN

17.13 Prompt: **Save the Format? [y]**

If you do not want to save the format,

PRESS:n

and go to 17.15.

If you want to save the format,

PRESS:RETURN

17.14 Prompt: **Enter format number -**

Any single-digit number or letter can be used as a format number. Two formats cannot be saved under the same number. Entering a previously used format number will erase the existing format stored under that number.

TYPE:<format number>

17.15 Prompt: **Enter field # to search - [1]**

If you do not want to print with the format, perform function **6. Escape from Function.**

If you want to print, you must run a search to select records for output.

If the data field in which you want to run your search is field #1,

PRESS:RETURN

If the data field in which you want to run your search is not field #1,

ENTER:<data field # for search>

17.16 Prompt: **Enter search text for field**
Prompt: **(press <left arrow> when done)**
On Screen: Search field open and close markers

ENTER:<search text>
PRESS:<left arrow>

For search procedures, see **35. Searches: Run.**

18. Labels: Format and Print pp.104-7

18.1 On Screen: **Main Menu**

Perform function **22. Printer: Setup.**

18.2 On Screen: **Search and Report menu**

PRESS:3

18.3 Prompt: **Use saved Format? [y]**

If no,

PRESS:n

and go to 18.5.

If yes,

PRESS:RETURN

18.4 Prompt: **Enter format number -**

TYPE:<Number of saved format to be used>

and go to 18.20.

18.5 Prompt: **Number of lines/label (9 max)? [3]**

If you want 3 lines per label,

PRESS:RETURN

If you do not want 3 lines per label,

ENTER:<number of lines per label>

up to a maximum of 9.

18.6 Prompt: **Field number for line 1? [1]**

If you want data field #1 to be printed in this position,

PRESS:RETURN

If you want another data field to be printed in this position,

ENTER:<field # of data field to be printed>

18.7 Prompt: **More fields in this line? [n]**

If no more fields in this line, and if this is the last line,

PRESS:RETURN

and go to 18.12.

If no more fields in this line, and if this is not the last line,

PRESS:RETURN

and go to 18.9.

If you want another field on this line,

PRESS:y

18.8 Prompt: **Field number for line 1? <previously entered #> [1]**

If the data field to be printed second in this line is field #1,

PRESS:RETURN

If the data field to be printed second in this line is not field #1,

ENTER:<data field # to be printed second in this line>

If there are no further lines in your label, go to 18.12.

If there is another line in your label, go to 18.9

18.9 Prompt: **Field number for line <next line #>? [1]**

If you want data field #1 to be printed in this position,

PRESS:RETURN

If you want another data field to be printed in this position,

ENTER:<field # of data field to be printed>

18.10 Prompt: **More fields in this line? [n]**

If no more fields in this line, and if this is the last line,

PRESS:RETURN

and go to 18.12.

If no more fields in this line, and if this is not the last line,

PRESS:RETURN

and go to 18.9.

If you want another field on this line,

PRESS:y

18.11 Prompt: **Field number for line <next line #>? <previously entered #> [1]**

If the data field to be printed second in this line is field #1,

PRESS:RETURN

If the data field to be printed second in this line is not field #1,

ENTER:<data field # to be printed second in this line>

If there are no further lines in your label, go to 18.12.

If there is another line in your label, go to 18.9.

18.12 Prompt: **Tab pos. of next label? [1]**

NOTE: If you want all possible data in label to be printed the tab position of next label must be equal to or greater than the total length of the fields in your longest line, plus one.

If the tab position of your next label is 1,

PRESS:RETURN

If the tab position of your next label is greater than 1,

ENTER:<tab position>

18.13 Prompt: **How many labels per row? [1]**

If you want one label per printed row,

PRESS:RETURN

If you want more than one label per printed row,

ENTER:<# of labels per row>

18.14 Prompt: **How many blank lines? [1]**

If you want one line between labels,

PRESS:RETURN

If you want more than one line between labels,

ENTER:<# of lines between labels>

18.15 Prompt: **All entries O.K.? [y]**

If you want to make changes in your format,

PRESS:n

and go to 18.3.

If all entries are correct,

PRESS:RETURN

18.16 Prompt: **Breakfield no.? [0]**

If you do not want a breakfield,

PRESS:RETURN

and go to 18.18.

If you want a breakfield,

ENTER:<breakfield #>

18.17 Prompt: **No. of chars. (10 max)? [3]**

If you want your breakpoint to occur with a change in the first three characters of the breakfield,

PRESS:RETURN

If you want your breakpoint to occur with a change in a different number of characters,

ENTER:<number of characters>

The maximum number of characters allowed is 10.

18.18 Prompt: **Save the Format? [y]**

If you don't want to save the format,

PRESS:n

and go to 18.20.

If you want to save the format,

PRESS:RETURN

18.19 Prompt: **Enter format number-**

Any single-digit number or letter can be used as a format number. Two formats cannot be saved under the same number. Entering a previously used format number will erase the existing format stored under that number.

TYPE:<format number>

18.20 **Enter field # to search - [1]**

If you do not want to print, perform function **6. Escape from Function.**

If you want to print, you must run a search to select the records for output.

If the data field in which you want to run your search is #1,

PRESS:RETURN

If the data field in which you want to run your search is not #1,

ENTER:<data field # for search>

18.21 Prompt: **Enter search text for field**
Prompt: **(press <left arrow> when done)**
On Screen: Search field open and close markers

ENTER:<search data>
PRESS:<left arrow>

For search procedures see **35. Searches: Run.** All printouts in this function are continuous. To abort printout at any time,

PRESS:<any key>

19. Menus: Display

19.1 Go to **25. Programs: Load** and load an ORACLE program.

19.2 On Screen: **Main Menu**

Prompts:
```
1 = Set data-file name.
2 = Create a new data file.
    - New record format
    - Modify existing format
3 = Modify an existing file.
    - Add a new record
    - Delete records
    - Change records
4 = Searching and Reporting.
    - Find records
    - Print reports
5 = Disk Utilities.
6 = Exit from this program.

  Enter selection -
```

19.3 Perform function **11. Filename: Set**

19.4 On Screen: **Main Menu**

To get the **Disk Utilities** menu

PRESS:5

Go to 19.6.

To get the **Search and Report** menu,

PRESS:4

19.5 On Screen: **Search and Report menu**

Prompt: **Select one of the following:**
```
1 = View Records
2 = Design a report
3 = Label printing
4 = Forms printing

     Utilities

5 = Setup Printer
6 = List fields
7 = Sort fields
8 = Exit to menu
9 = Set password (Secured program only.)

   Enter Selection -
```

To return to **Main Menu,**

PRESS:8

Go to 19.4.

19.6 On Screen: **Disk Utilities menu**
Prompt: **Select one of the following:**
1 = Show all files
2 = Change disk unit
3 = Scratch files
4 = Rename files
5 = Format disk
6 = Copy a disk
7 = Exit to menu

Enter Selection -

To return to **Main Menu,**

PRESS:7

and go to 19.4.

20. **Passwords: Change** pp.133-5

To change previously-set passwords you must re-create your secured program, incorporating the new passwords. Perform function **37. Secured Program: Create.**

21. **Passwords: Enter** pp.133-5

In secured files, passwords must be entered for all options in which data is displayed, printed, or modified.

21.1 In order for passwords to be entered, the secured ORACLE program with the proper passwords set must be loaded.

If a secured program has been saved, perform function **25. Programs: Load.**

If no secured program has been saved, perform function **37. Secured Program: Create.**

21.2 On Screen: **Main Menu**

Perform function **11. Filename: Set.**

21.3 On Screen: **Main Menu**

To enter password to enter the **Modify a File** mode,

PRESS:3

and go to 21.4.

To enter password to enter **Search and Report** options,

PRESS:4

and go to 21.5.

21.4 Prompt: **Enter your Password -**

Only the master code gives access to this mode.

ENTER:<4-digit master code>

The characters of the code will not appear on the screen as you type them. After you enter the code, the **File Maintenance** screen is displayed.

21.5 On Screen: **Search and Report** menu

PRESS:9

Prompt: **Enter your Password -**

ENTER:<code - level 1, 2, or 3>

The characters of the code will not appear on the screen as you type them. After you've entered the code, the **Search and Report** menu is displayed.

22. Printer: Setup pp.71-3

22.1 On Screen: **Main Menu**

PRESS:4

22.2 On Screen: **Search and Report** menu

PRESS:5

22.3 Prompt: **Enter number of linefeeds - [0]**

If you want single-spacing, and your printer has automatic linefeed,

PRESS:RETURN

If you want single-spacing, and your printer does not have automatic linefeed,

ENTER:1

For double-spacing, add 1 to the above entries; for triple-spacing, add 2; etc.

22.4 Prompt: **Printer device number - [4]**

If your printer device number is 4,

PRESS:RETURN

If your printer device number is not 4,

ENTER:<printer device number>

22.5 Prompt: **Output? (Screen or Printer) - [s]**

PRESS:p

22.6 Prompt: **Enter page header or RETURN**

If you want a page header and automatic page numbering,

TYPE:<header text and appropriate spacing>
ENTER:#

If you want a page header with no page numbering,

ENTER:<header text>

If you do not want a page header,

PRESS:RETURN

22.7 Prompt: **Lines/page? [55]**

If you want 55 lines per page,

PRESS:RETURN

If you do not want 55 lines per page,

ENTER:<# of lines per page>

22.8 Prompt: **Ascii or Cbm Code**

If you want Ascii code,

PRESS:a

If you want Cbm code,

PRESS:c

22.9 On Screen: **Search and Report menu**

Printer setup is complete.

23. Printer: Shut off pp.71-3; p.74

23.1 On Screen: **Main Menu**

PRESS:4

23.2 On Screen: **Search and Report** menu

PRESS:5

23.3 Prompt: **Enter number of line feeds - [0]**

PRESS:RETURN

23.4 Prompt:**Printer device number - [4]**

PRESS:RETURN

23.5 Prompt: **Output? (Screen or Printer) - [s]**

PRESS:RETURN

24. Programs: Backup pp.14-15

24.1 If you have a dual drive, perform function **2. Disk: Copy,** and copy your program disk. This will backup all programs on the disk.

If you have a single drive go to 24.2.

24.2 If you do not have a formatted disk, perform function **5. Disk: Format.**

If you have a formatted disk ready, go to 24.3.

24.3 Perform function **25. Programs: Load** and load, but do not run (steps 25.1 to 25.2) the program to be copied.

24.4 When the flashing cursor returns to the screen, remove the original disk from the drive, and insert the newly formatted disk.

ENTER:save"<program name>",8

The backup is complete when the **ready** signal and flashing cursor return to the screen.

25. Programs: Load pp.16-17

25.1 If your computer is not in BASIC perform function **7. Exit to BASIC.**

25.2 Insert the disk containing the program to be loaded into drive 0 (zero).

ENTER:load"<program name>",8

25.3 When the **ready** signal and flashing cursor return to the screen,

ENTER:run

26. Records: Add from Keyboard pp.42-52

26.1 On Screen: **Main Menu**

PRESS:3

26.2 On Screen: **File Maintenance** screen, first page of record format.
Prompt: **Enter Exit, Add, or Update -**

PRESS:a

26.3 Prompt: **File or Keyboard? [k]**

PRESS:RETURN

26.4 On Screen: **ADD RECORD** screen, with cursor in field.
Prompt: **Enter record data (<left arrow> key to finish)**

If there is data to be entered in this field,

TYPE:<data>

and

PRESS:RETURN

to get to the next field.

If there is no data to be entered in this field,

PRESS:RETURN

to get to the next field.

26.5 On Screen: Cursor in next field.
Prompt: **Enter record data (<left arrow> key to finish)**

If there is more data to be entered on this page, go to 26.4.

If there is no more data to be entered on this page, go to 26.6.

26.6 On Screen: All necessary data entered.
Prompt: **Enter record data (<left arrow> key to finish)**

PRESS:<left arrow>

If this is the last page of your screen, go to 26.8.

26.7 On Screen: Next page of record, with cursor in field.

Go to 26.4.

26.8 Prompt: **Add another record (y/n)? [y]**

If you want to add another record,

PRESS:RETURN

and go to 26.4.

If you do not want to add another record,

PRESS:n

26.9 On Screen: **File Maintenance** screen, first page of record format.
Prompt: **Enter Exit, Add, or Update**

To add more records,

PRESS:a

and go to 26.3.

To exit to **Main Menu,**

PRESS:e

27. Records: Add from Sequential File pp.122-4

27.1 On Screen: **Main Menu**

PRESS:3

27.2 On Screen: **File Maintenance** screen, first page of record format.
Prompt: **Enter Exit, Add, or Update**

PRESS:a

27.3 Prompt: **File or Keyboard? [k]**

PRESS:f

27.4 Prompt: **Enter Filename?**

ENTER:<name of sequential file>

ORACLE will now write all the data from the sequential file into the data file.

27.5 On Screen: **File Maintenance** screen, first page of record format.
Prompt: **Enter Exit, Add, or Update**

To add more records,

PRESS:a

and go to 27.3.

To exit to **Main Menu**,

PRESS:e

28. Records: Change pp.65-70

28.1 On Screen: **Main Menu**

PRESS:3

28.2 On Screen: **File Maintenance** screen, first page of record format
Prompt: **Enter Exit, Add, or Update -**

PRESS:u

28.3 On Screen: **UPDATE RECORD** screen, first page of record format
Prompt: **Enter search data (press <left arrow> when done)**

To run search perform function **35. Searches: Run.**

28.4 On Screen: First page of selected record.
Prompt: **Change,Del,Next,Prev,Match,Rec.#,Exit?**

PRESS:c

28.5 On Screen: Cursor in screen field.
Prompt: **Enter changes (press <left arrow> when done)**

If there are changes in this field,

TYPE:<data>
PRESS:RETURN

If there are no changes in this field,

PRESS:RETURN

28.6 On Screen: Cursor in next field.
Prompt: **Enter changes (press <left arrow> when done)**

If there are more changes to be made, go to 28.5.

If there are no more changes to be made, go to 28.7.

28.7 On Screen: All changes correctly entered.
Prompt: **Enter changes (press <left arrow> when done)**

PRESS:<left arrow>

If this was the last page of your record go to 28.9.

28.8 On Screen: Next page of record.

Go to 28.5.

28.9 On Screen: First page of changed record.
Prompt: **Change,Del,Next,Prev,Match,Rec.#,Exit?**

To make further changes,

PRESS:c

To search for further records to update,

PRESS:<n, p, m, or r>

To exit to **Main Menu,**

PRESS:e

29. Records: Delete pp.65-70

29.1 On Screen: **Main Menu**

PRESS:3

29.2 On Screen: **File Maintenance** screen, first page of record format
Prompt: **Enter Exit, Add, or Update -**

PRESS:u

29.3 On Screen: **UPDATE RECORD** screen, first page of record format
Prompt: **Enter search data (press <left arrow> when done)**

To run search perform function **35. Searches: Run.**

29.4 On Screen: First page of selected record.
Prompt: **Change,Del,Next,Prev,Match,Rec.#,Exit?**

PRESS:d

29.5 Prompt: **Deletion - are you sure? [n]**

If you do not want to delete this record,

PRESS:RETURN

To delete record,

PRESS:y

29.6 On Screen: **UPDATE RECORD** screen, first page of record format
Prompt: **Next,Prev,Match,Rec.#,Exit?**

To search for further records to update,

PRESS:<n,p,m, or r>

To exit to **Main Menu,**

PRESS:e

30. Records: Print p.73

30.1 On Screen: **Main Menu**

Perform function **22. Printer: Setup.**

30.2 On Screen: **Search and Report** menu

PRESS:1

30.3 On Screen: Page of record format
Prompt: Enter search data (press <left arrow> when done)

Perform function **35. Searches: Run.**

30.4 On Screen: First page of selected record.
Prompt: Exit,Next,Prev,Match, or Rec.#?

To continue the search,

PRESS:< n, p, m, or r>

and go to 30.4.

To abort search,

PRESS:e

and go to 30.8.

To accept the record,

PRESS:<any key except e, n, p, m, or r>

30.5 **Prompt: Is this the record you want (y/n)? [y]**

If this is not the record you want,

PRESS:n

and go to 30.4.

If this is the record you want,

PRESS:RETURN

30.6 **Prompt: Print this Screen? [n]**

If you want to print,

PRESS:y

The screen will be printed.

If you do not want to print,

PRESS:RETURN

In either case, if this is the last page of your record, go to 30.8.

30.7 Prompt: **Do you want the next page? [y]**

If you want the next page or subsequent pages,

PRESS:RETURN

The next page of the record appears on the screen. Go to 30.6.

If you do not want the next page or subsequent pages,

PRESS:n

30.8 On Screen: **Search and Report** menu

To print more records,

PRESS:1

31. Records: View pp.54-64

31.1 On Screen: **Main Menu**

PRESS:4

31.2 On Screen: **Search and Report** menu

PRESS:1

31.3 On Screen: First page of record format.
Prompt: **Enter search data (press <left arrow> when done)**

Perform function **35. Searches: Run.**

31.4 On Screen: First page of selected record.
Prompt: **Exit,Next,Prev,Match, or Rec.#?**

To continue the search,

PRESS:< n, p, m, or r>

and go to 31.4.

To abort the search,

PRESS:e

and go to 31.7.

To accept the record,

PRESS:<any key except e, n, p, m, or r>

31.5 Prompt: **Is this the record you want (y/n)? [y]**

If this is not the record you want,

PRESS:n

and go to 31.4.

If this is the record you want, do not respond to the prompt until you are finished with it. Then

PRESS:RETURN

If there are no more pages in your record, you are returned to the **Search and Report** menu. Go to 31.7.

31.6 Prompt: **Do you want the next page? [y]**

If yes,

PRESS:RETURN

and go to 31.6.

If no,

PRESS:n

31.7 On Screen: **Search and Report menu**

To view another record,

PRESS:1

and go to 31.3.

32. Reports: Format and Print pp.74-87

32.1 On Screen: **Main Menu**

PRESS:4

32.2 On Screen: **Search and Report** menu

If reports are to be printed, perform function **22. Printer: Setup.**

32.3 On Screen: **Search and Report menu**

PRESS:2

32.4 Prompt: **Use a saved Format? [y]**

If you do not want to use a saved format,

PRESS:n

and go to 32.7.

If you want to use a saved format,

PRESS:RETURN

32.5 Prompt: **Enter format number -**

TYPE:<number of saved format>

32.6 On Screen: Saved Format
Prompt: **Any changes? [y]**

If there are no changes to be made,

PRESS:n

and go to 32.17.

If there are changes to be made,

PRESS:RETURN

32.7 On Screen: Formatting screen or saved format with field symbols displayed.
Prompt: **Enter format (<up arrow>=start field, <left arrow>=quit)**

Only when this prompt is displayed can out-of-field characters be typed, or full screen editing be used to shorten, lengthen, or move existing screen displays.

To create a format field, move your cursor to the position you want the field to begin in and

PRESS:<up arrow>

32.8 On Screen: Open Field Marker
Prompt: **Enter field # for position -**

If the field is to be a variable field, go to 32.12.

If the field is to be a data field,

ENTER:<data field number>

and go to 32.9.

If the field is to be a non-printing search field,

ENTER:<128 + data field number for search>

32.9 Prompt: **<field #>=<field name> len=xxx This one O.K.? [y]**

If this is the data field you want,

PRESS:RETURN

and go to 32.11.

If this is not the data field you want,

PRESS:n

32.10 Prompt: **<next field #>=<next field name> len=xxx This one O.K.? [y]**

If this is not the data field you want,

PRESS:n

and go to 32.10.

If this is the data field you want,

PRESS:RETURN

32.11 On Screen: Close field marker, end of field symbol.
Prompt: **Enter format (<up arrow>=start field, <left arrow>=quit)**

If you want another field in the format, go to 32.7.

If you do not want another field,

PRESS:<left arrow>

and go to 32.14.

32.12 On Screen: Open Field Marker
Prompt: **Enter field # for position -**

If the variable field is to be the first breakfield created on this screen, perform function **1. Breakpoint: Specify.** Then go to 32.11.

If the variable field is anything but the first breakfield,

ENTER:<variable field number: 0 or 100 - 115>

32.13 On Screen: Open field marker with flashing cursor.
Prompt: **Enter variable field data & RETURN**

ENTER:<variable field data>

Go to 32.11.

32.14 Prompt: **Any Changes? [n]**

If there are changes to be made to the format,

PRESS:y

and go to 32.7.

If there are no changes to be made to the format,

PRESS:RETURN

32.15 Prompt: **Save the format? [y]**

If you don't want to save the format,

PRESS:n

and go to 32.17.

If you want to save the format,

PRESS:RETURN

32.16 Prompt: **Enter format number -**

Any single-digit number or letter can be used as a format number. However, two formats cannot be saved under the same number. Entering a previously used format number will erase the existing format stored under that number.

TYPE:<format number>

32.17 On Screen: Cursor in data field.
Prompt: **Enter search data (press <left arrow> when done)**

If you do not want to output data to screen or printer at this time, perform function **6. Escape from Function.**

If you want to output data, you must run a search. For search procedures see **35. Searches: Run.**

If printer is set up, printout will begin when the search has begun.

33. Screen: Print pp.72-3

Any screen with a cursor flashing in a field can be printed if the printer is set up.

33.1 To set up printer, perform function **22. Printer: Setup.**

33.2 On Screen: Cursor flashing in field.

To print screen,

PRESS:"

34. Searches: Interrupt p.82

34.1 During continuous output, or if the **Searching** prompt is on the screen, any search can be interrupted. To interrupt a search,

PRESS:<any key>

34.2 Prompt: **Interrupt? [y]**

If you do not want to interrupt the search,

PRESS:n

The search will continue from where it was broken.

If you do want to interrupt the search,

PRESS:RETURN

34.3 On Screen: Search prompts appropriate to function, and (except in **Label Printing** and **Forms Printing**) the R# of record reached in search.

You may now exit the search or continue, according to the prompt displayed.

35. Searches: Run pp.54-64

Searches can be run in the **Update Records**, **View Records**, **Design a Report**, **Label Printing**, and **Forms Printing** modes.

Searches can only be run when the cursor is flashing in a data field on the screen, and when the prompt reads: **Enter search data (press <left arrow> when done).**

35.1 On Screen: Cursor flashing in data field.
Prompt: **Enter search data (press <left arrow> when done)**

Move cursor to the field in which you want to search.

35.2 On Screen: Cursor flashing in correct data field.
Prompt: **Enter search data (press <left arrow> when done)**

For an 'unequal match' search, go to 35.3.

For a 'string' or 'wild card' search, go to 35.5.

For an 'any match' search, go to 35.6.

35.3 On Screen: Cursor flashing in correct data field.
Prompt: **Enter search data (press <left arrow> when done)**

If you want a 'greater than' search,

TYPE:>

If you want a 'less than' search,

TYPE:<

35.4 On Screen: 'Greater than' or 'less than' symbol on top line. Cursor flashing in correct data field.
Prompt: **Enter search data (press <left arrow> when done)**

TYPE:<text string for search>

Search strings must be typed into search fields properly right- or left-justified.

If you want a second search condition, move cursor to second search field and go to 35.7.

If you do not want a second search condition, or if you have already specified two search conditions, or if you are in the **Label Printing** or **Forms Printing** mode, go to 35.8.

35.5 On Screen: Cursor flashing in correct data field.
Prompt: **Enter search data (press <left arrow> when done)**

Search strings must be typed into search fields properly right- or left-justified.

TYPE:<text string for search>

If you want a second search condition, move cursor to second search field and go to 35.7.

If you do not want a second search condition, or if you have already specified two search conditions, or if you are in the **Label Printing** or **Forms Printing** mode, go to 35.8.

35.6 On Screen: Cursor flashing in correct data field.
Prompt: **Enter search data (press <left arrow> when done)**

TYPE:*

into the first position in the search field. Go to 35.8.

35.7 On Screen: Cursor flashing in second search field.
Prompt: **Enter search data (press <left arrow> when done)**

To specify a second search condition,

TYPE:+

A '+' sign will appear on the top line of your screen. Go to 35.2.

35.8 On Screen: All search data entered.
Prompt: **Enter search data (press <left arrow> when done)**

PRESS:<left arrow>

If you are in the **Update Records** mode, go to 35.9.

If you are in the **View Records** mode, go to 35.10.

If you are in the **Design a Report** mode, go to 35.11.

If you are in the **Forms Printing** mode, go to 35.12.

If you are in the **Label Printing** mode, go to 35.14.

35.9 On Screen: Selected record.
Prompt: **Change,Del,Next,Prev,Match,Rec.#,Exit?**

If this is the record you want to change or delete,

PRESS:<c or d>

To select next record,

PRESS:n

To select previous record,

PRESS:p

To select next match,

PRESS:m

To select a specific record number,

PRESS:r

and then

ENTER:<record number>

To exit mode,

PRESS:e

35.10 On Screen: Selected record.
Prompt: **Exit,Next,Prev,Match or Rec.# ?**

To discontinue search,

PRESS:<any key except e, n, p, m, or r>

To exit function,

PRESS:e

For next record,

PRESS:n

For previous record,

PRESS:p

For next match,

PRESS:m

For a specific record number,

PRESS:r

and then

ENTER:<record number>

35.11 On Screen: Selected record.
Prompt: **Enter q(quit), n(next match), c(cont.)**

If your printer is set up, the selected record will have printed.

To exit function,

PRESS:q

To select next match,

PRESS:n

For continuous output,

PRESS:c

35.12 On Screen: Search data in field.
Prompt: **Enter q(quit), n(next match), c(cont.)**

If your printer is set up, the selected record will have printed.

To exit function,

PRESS:q

To select next match,

PRESS:n

For continuous output,

PRESS:c

35.13 On Screen: Search data in field.

If your printer is set up, continuous printout has begun.

36. Secured File: Create pp.133-5

36.1 Load secured program.

For information on how to create a secured program, see **37. Secured Program: Create.**

For information on how to load the program, see **25. Programs: Load.**

36.2 Go to **13. Files: Create.** Complete steps 13.1 to 13.4.

36.3 Prompt: **Enter security level (1-3) or RETURN**

To create a security 'level 3' field,

PRESS:3

To create a security 'level 2' field,

PRESS:2

To create a security 'level 1' field,

PRESS:1

To create an unsecured field,

PRESS:RETURN

Go back to **13. Files: Create.** Continue with step 13.5.

37. Secured Program: Create pp.133-5

37.1 If your computer is not in BASIC, perform function **7. Exit to BASIC.**

37.2 If you are using a Commodore 64, go to 37.3.

If you are not using a Commodore 64, go to 37.4.

37.3 **ENTER:**load"Oracle 64.sec",8

When the **ready** signal and flashing cursor reappear,

ENTER:run

Go to 37.5.

37.4 Load ORACLE program.

37.5 Prompt: **Do you want security codes? [n]**

If you want security codes,

PRESS:y

If you do not want security codes,

PRESS:RETURN

37.6 Prompt: **Do you have a dual disk drive? [n]**

If you do not have a dual disk drive,

PRESS:RETURN

If you do have a dual disk drive,

PRESS:y

If you did not ask for security codes, go to 37.10.

37.7 Prompt: **Enter Master code (4 char)**

TYPE:<any 4 characters except xxxx>

37.8 Prompt: **Enter level 1 (3 char)**

TYPE:<any 3 characters>

37.9 Prompt: **Enter level 2 (3 char)**

TYPE:<any 3 characters>

If 'level 1' and 'level 2' security are to be distinct, their codes must be different.

37.10 Prompt: **Save Secured Program**

If you do not want to save the secured program, go to 37.11.

If you want to save the secured program, remove your program disk from the drive, insert a formatted disk, and

ENTER:save"<drive #>:<name of secured database>",8

38. Sequential File: Create pp.122-4

38.1 Perform function **22. Printer: Setup.** Complete steps 22.1 to 22.3.

38.2 Prompt: **Printer device number - [4]**

ENTER:<disk drive device #>

38.3 Prompt: **Enter Filename?**

ENTER:<filename for sequential file>

38.4 Prompt: **Ascii or Cbm Code?**

PRESS:c

38.5 Prompt: **Do you want all spaces? [n]**

If you want extra spaces suppressed,

PRESS:RETURN

If you want all spaces,

PRESS:y

38.6 On Screen: **Search and Report menu**

Go to **17. Forms: Format and Print** or **32. Reports: Format and Print**, and create a format for your sequential file, creating one field for each data field you want written into the sequential file. Sequential files that are to contain fixed-text fields must be created in the **Design Reports** mode.

For detailed information on sequential files, see pp.122-4

Section 2 - Appendices

Appendix I - Inserting Your ROM Chip Page 201
Appendix II - ORACLE Special Keys Page 202
Appendix III - Sort Chart Page 203
Appendix IV - Security Access Chart Page 204
Appendix V - ORACLE Field Types Page 205
Appendix VI - Error Messages and Conditions Page 207
Appendix VII - .HDR and .REL File Layout Page 209
Appendix VIII - Resetting ORACLE Defaults Page 211

Appendix I - Inserting Your ROM Chip

1) Turn off your computer and disconnect it from its power supply.

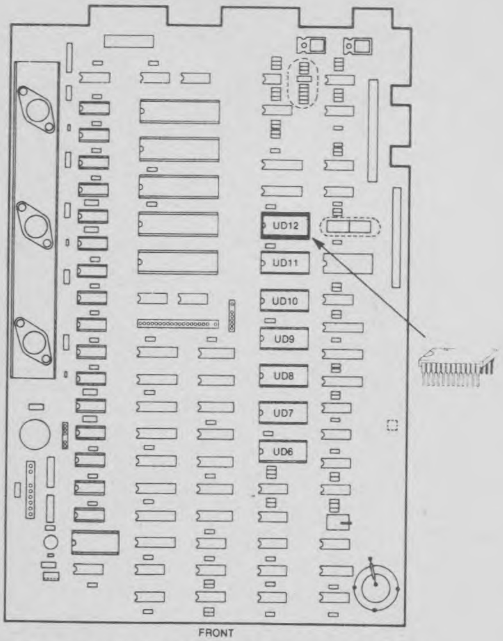
2) Remove the screws holding down the computer's housing, and prop the housing open. (See your computer manual for instructions.)

3) Consult the diagram below to find the right socket for the chip in your computer. The socket is #UD3 in model 2001, and #UD12 in other models.

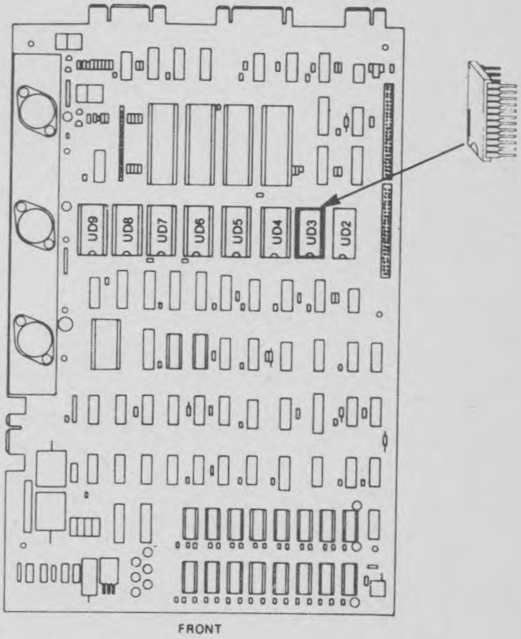
4) Remove the ROM chip from its package. Making sure that the notch on the chip is facing the same way as shown below, firmly seat the chip in its socket. Make sure all the pins are correctly inserted.

   CAUTION: ROM chips are very delicate. Handle them with care. Take special care not to bend the pins.

5) Close the computer housing and refasten it.



8032 SERIES PRINTED CIRCUIT BOARD



2001 SERIES PRINTED CIRCUIT BOARD

Appendix II - ORACLE Special Keys

Outside the normal keyboard characters several special keys are used in ORACLE. This table identifies these keys and their functions in ORACLE.

KEY 8032/4032/2001	 Commodore 64	**FUNCTION**
<up arrow>	<up arrow>	1) start a field 2) copy data from previous record (**Add Records** mode) 3) set direct printer command
<left arrow>	<left arrow>	1) run a search 2) store format on screen in memory
<back slash>	<Brit. pound>	solid screen underline
(shift)RUN/STOP	f8	escape to **Main Menu**
"	"	print screen
RUN/STOP	RUN/STOP	turn line 'insert/ delete' function on
+	+	insert line
-	-	delete line
	f1	change background color
	f2	change foreground color
	f3	change character color
@	@	first character in variable fields with calculations
*	*	'any match' search
?	?	1) 'wild card' search 2) first character in variable field for data entry at each record

Appendix III - Sort Chart

The following table lists the number of records you can sort at various depths within ORACLE on the 8032/4032 and on the Commodore 64. The last column lists the number of sorts you can do using the SUPERSORT program.

With each subsort the number of sorts possible is reduced. The depth of the main sort and all subsorts should be totalled to calculate the number of records sortable.

Depth	8032/4032	C-64	SUPERSORT
3	2150	4262	5866
4	1792	3552	4889
5	1536	3044	4190
6	1344	2664	3666
7	1194	2358	3259
8	1075	2131	2933
9	977	1937	2666
10	896	1776	2444
11	827	1639	2256
12	768	1522	2095
13	716	1420	1955
14	672	1352	1833
15	632	1252	1725

Appendix IV - Security Access Chart

The following chart summarizes the access given to a fully secured file with different level codes.

CODE	DISPLAY/PRINT FUNCTIONS Field security level: 3	 2	 1	 0	FILE MAINTENANCE FUNCTIONS
MASTER	yes	yes	yes	yes	yes
LEVEL 2	no	yes	yes	yes	no
LEVEL 1	no	no	yes	yes	no
NO CODE	no	no	no	yes	no

Appendix V - ORACLE Field Types

The **Design a Report** mode uses two basic field types -- data fields and variable fields. Data fields are numbered according to their field number in the record format. Variable fields are numbered according to where in a report they print, if they print at all.

Field #	
0	Variable field which prints in the body of a report.
1-99	Data fields.
100	Variable field which prints only at the beginning of a report.
101	Variable field which prints only at the beginning of each page of a report (page header must be set).
102	Variable field which prints only at the end of a report.
103	Variable field which prints only at the end of each page of a report. (Page header must be set.)
104-113	Variable fields which print only at the bottom of the page (page header must be set) and which clear accumulators 'a' through 'j', respectively.
114	Non-printing variable field.
115	Breakfield. A breakfield is a variable field which prints only at a breakpoint, and whose first use in a format allows breakpoints to be defined. Breakfields clear accumulators 'a' through 'm' at breakpoints if the sum function using the accumulator is entered in the breakfield as its variable field data.
129-227	Non-printing data fields. Used to run searches in data fields whose contents are not to be printed. The number for a non-printing data field is obtained by adding 128 to the data field number.

The function of variable fields depends on the data which is entered in them:

- '@' in the first position in the field means that the variable field is a calculation field.
- '?' in the first position means that data output will stop at this field in every record so that text can be entered from the keyboard.
- Any other character in the first position means that the variable field contains fixed text.
- A blank variable alone on a line forces a blank line in printouts.
- Variable fields may contain direct printer commands, or the ';' command.

During the creation of report formats, various symbols follow the close field marker on the screen. ORACLE uses these symbols internally to differentiate among field numbers. You may find these symbols useful for screen editing your report formats. The number of a field is the ASCII code for the symbol. For information on the ASCII sequence see your computer manual.

Appendix VI - ORACLE Error Messages and Conditions

An error message may be displayed with or without a flashing cursor on the screen. If there is no flashing cursor

PRESS:<any key>

and you will be returned to the menu. If there is a flashing cursor on the screen, you have the screen editing capabilities to correct the error -- once the error is corrected the message will disappear.

End of file (press any key)

Not an error: indicates that the end of the data file has been reached.

Error - Field exceeds page

There is not enough space on the page for a field of the specified length.

Error - No Fields Specified

You cannot create a page with no fields on it.

Error - No such field

A field number has been specified that is not currently defined.

Error - 'open' & 'close' unmatched

The number of open markers and the number of close markers do not agree. The cursor flashes where two identical markers appear consecutively or, if the last close marker is missing, on the bottom line.

Error - too many fields

More than 99 fields have been used. The number of fields must be reduced to 99 or fewer.

Format error - please retype number

The number entered is too large for the remaining spaces on the screen.

Format file error

The specified format file is not found in the disk directory.

Out-of-sort memory - creating unsorted file

The key fields for all records in the file exceed available memory. Reduce depth of sort or use SUPERSORT program to sort.

Record too big for memory

The maximum record size has been exceeded. The size must be reduced.

Syntax error - strike any key

A mathematical formula cannot be evaluated. Check that all formulas are typed correctly.

There are two error conditions which may occur in ORACLE, but which are not announced by screen messages. However, the nature of the error can be read from the screen. Both conditions occur in the **Design a Report** mode.

i) If a formula field fills with question marks during output, it means that the field is not long enough to contain the results of the calculation.

ii) If a data field is filled with the wrong contents, or if it is filled with random characters, it means that the field symbol which follows the close field marker has been erased, or erased and retyped incorrectly.

Appendix VII - Technical Note: .HDR and .REL File Layout

The following information will enable the ORACLE user to interface the data files of the system with other programs.

.REL (ORACLE .REL files are relative files)

This file is a random access file of fixed length where data fields are strung together one by one. It may occupy more than one disk record per user record if the specified length exceeds 254 characters. It is formatted as follows:

Record 1 points to the first available record to be added to the file using a 2-byte (character) pointer where the first byte is the low half of the number (mod 256) and the second byte is the high half. For example, if record 300 is the next available empty record in the file, then record 1 will contain the characters '44', '1', and '13'. The last byte is a record terminator (RETURN) character.

Records 2 through 299 will be normal data records, and if record 300 is not the end of the file then it will contain the pointer to the next available blank record. If it is the end of the file then record 300 will contain the bytes '254', '0', and '0'. The character '254' is the empty record indicator. The pointers ('254' and two bytes) of the empty records are not necessarily in any order, since their value is determined if a record is randomly deleted. It is thus possible that the pointer will point backwards in the file. For example, in the above case, record 300 may have pointed to record 150. The data in the file is written in CBM display format; that is a=1, b=2, A=65, B=66, etc. Conversion algorithms may be found in the Commodore reference manual for BASIC 4.

.HDR (ORACLE .HDR files are sequential files)

This file contains all the formatting information of the record layout. The length of the record, its blocking into disk records, the number of pages, all field names, types, lengths, and all screen comments are stored in this file. The layout of the bytes (characters) has the following meaning:

BYTE	RANGE	DESCRIPTION
01	0-255	Record length, low
02	0-40	Record length, high
03	0-99	Number of fields in record
04	0-9	Number of pages in record
05	0-45	Number of physical disk records
06	1-253	Their length
07-10	0	Spare bytes (4)
11	13	Carriage return character
12	1-9	Page this field is on
13	0-255	Screen address of field, low
14	128-255	Screen address of field, high
15	0-255	Number of leading spaces
16	0-127	First ASCII character of field name or description
17-nn	0-127	All other characters of field name (terminated by a null '0' char.)
nn+1	0-255	Length of this field, low
nn+2	0-8	Length of this field, high (if '128' then field is remark only)
nn+3	0-55	Attribute of this field
nn+4	13	Carriage return character
nn+5		Repeat from position 12

Appendix VIII - Resetting ORACLE Defaults

It is possible to set ORACLE default values for sort depths, disk unit, disk drive, and disk type.

When the **save secured program** prompt is displayed after loading, use the following BASIC 'poke' commands to enter the values desired.

	8032	4032/2001	C-64
SORT DEPTH	1100	1110	2160
DISK UNIT	1068	1078	2128
DISK TYPE	1095	1105	n/a
DISK DRIVE	1072	1082	see * below

*To set disk drive default for C-64 poke 2124,202 for 0; poke 2124,204 for 1.

Section 3 - Glossary

accumulator	A storage location in your computer's memory where numbers are added up and the result is stored. ORACLE uses 36 accumulators, designated 'a' through 'z', and '0' through '9'.
alpha field	A field that is given the alpha attribute during file creation. It will accept as data only alphabetic characters, asterisks, and question marks.
any match search	A search using '*' as the search data, which selects all records from a file.
attribute	A characteristic of a field. There are three basic types of attributed fields -- alpha fields, numeric fields, and key fields.
backup	A copy of some or all of the data on a disk written onto another disk. Usually, when a backup is of a whole disk, the disk onto which the data is copied is formatted identically to the disk from which it is copied.
BASIC	An acronym for 'Beginner's All-purpose Symbolic Instruction Code', a conversational language for writing programs on computers.
block	Two hundred and fifty-six bytes.
breakfield	A variable field which prints only at a breakpoint. ORACLE breakfields are 115-type fields.

breakpoint	A point in data output at which the normal sequence of output is broken so that breakfields can print. Breakpoints must be defined whenever breakfields are used in a report format.
byte	The amount of space in your computer's memory used to store a single character.
close field marker	A graphic character which marks the end of a field on the screen.
cursor	The reverse video rectangle which marks the location where a character typed on the keyboard will appear on the screen.
database	An electronic filing system for holding information.
data field	A field in a record format or report format in which field information is entered.
default	1) A value or response assumed by the computer when no other value or response is specified by the user. 2) To accept a default value. In ORACLE you always default by pressing the RETURN key.
rect printer command	A command issued to your printer from within ORACLE, but using the printer's own codes.
error message	A message that appears on the screen, indicating an error in entering information or an error in the system.
field	A sub-unit of a record. A part of a record that contains a specific piece of information.

field number	For a data field, the field number is its field position number in the record format. For a variable field, the field number is the field type number.
field position number	A number assigned to fields according to their screen position by counting in sequence from left to right and top to bottom.
field type	A number assigned to variable fields according to where they print in a printout and which accumulator they clear. ORACLE has variable field types 0, and 100 to 115.
file	A collection of electronically stored information.
floppy disk	A magnetic disk on which electronic files and programs are stored.
force page	A command which tells your printer to stop printing on the current page and move to the next page, even though the current page is not full.
formatting a disk	The process by which a disk is given a name and a two-digit identification number by which your computer can recognize it.
hard copy	Data printed out on paper.
header	A line of characters which is printed only on the first line of each page of a printout.
key	1) A small device which plugs into the control port of your Commodore 64, and which allows you to load the ORACLE program. 2) The attribute of a sorted field.

key field	A field which is automatically sorted by ORACLE, and for which a key file is written.
key file	A special file created by ORACLE, in which it stores and sorts the contents of a key field.
left-justification	The positioning of data within a field as far to the left as possible.
linefeed	A signal sent by the computer to create an automatic carriage return at the end of a printed line.
logical-and search	A search which uses two search conditions at once.
menu	A collection of screen prompts for different ORACLE functions. ORACLE has 3 menus -- the **Main Menu,** the **Search and Report** menu, and the **Disk Utilities** menu.
numeric field	A field which accepts only numbers, asterisks, question marks, hyphens, periods, and backslashes.
open field marker	A graphic character which marks the beginning of a field on the screen.
ut-of-field character	A character which appears on a record format screen or on a report format screen, but which is not contained in a field.
page	A sub-unit of a record. A collection of fields in a record which appear on a single screen.

prompt	A message that appears on the screen advising the user of the next step in a procedure.
read	To transfer data from any device (usually disk or keyboard) into the computer's memory.
record	A sub-unit of a file. All records in a given file are formatted identically.
record format	The layout of a record -- i.e., the position, length, and attribute of each of its fields.
relative file	A file that is structured so that any individual piece of data can be found directly (often referred to simply as 'file').
report format	The layout of a report -- the position, length, and type of each of its fields.
reverse video	The switching of background and foreground colors on the screen.
ight-justification	The positioning of data within a field as far to the right as possible.
ROM chip	ROM stands for 'read only memory'. With any model other than the Commodore 64, a special ROM chip must be installed in your computer in order to load ORACLE.
scratch	To delete a file permanently from a disk.
sequential file	A file that is unstructured, so that an individual piece of data in it can be found only by running through all its data in sequence.

sort	To arrange the records in a file in the alphabetic or numeric order of the contents of one of its fields.
string	A sequence of characters or spaces.
string search	A search in which the search data is a string of characters.
subfile	A file which is formatted exactly the same as another file, and which contains some, but not all, of its records.
unequal match search	A search that looks for all records that contain, in a given field, a numeric or alphabetic value either greater than or less than a given value.
variable field	Any field in a report format that is not a data field, so called because its function and data can vary. In ORACLE variable fields are identified by the numbers 0 and 100-115.
wild card search	A string search which will select records with any character in the positions which contain the '?' character in the string.
workdisk	A disk on which data files are saved.
write	To save a file on a disk.
write protect tab	A small tab fitted over the notch on the cardboard case of a floppy disk, which makes it impossible to write on the disk.

accumulator 94,143,146,150-1
adding data 42-52,180-2
backup disks 14,121,155
backup programs 14-15,179
BASIC 9,139
  exit to BASIC 118,157
blocks 38,137,139
breakfields 145-6,151
breakpoint 146,155
calculations 93-103,140,149-51
  accumulator 94,143,146,150-1
  decimals 99-100,149
  match count 95,151
  sums 94,146,150-1
defaults 35
  resetting defaults 211
delete records 67-8,183-4
direct printer commands 142
disk drive 7,119-20,127-8,156,157
disks 10, 137
  copying 14,121,155
  formatting 15,21-2,121,156
  ID numbers 15,22
  workdisk 21
error messages 19,35,39,98,129,131,150,207-8
escape from function 19-20,157
field list 41,73,158
field markers 27,28,35,75,76,85,140
field number 75,83
field position number 94,140,149
field symbols 75,76,84
field types 205-6
  alpha 27,44,139
  breakfields 145-6,151
  data 25-37,75,82
  key 27,63,64,69-70,115,119,139
  non-printing 85,93-4
  numeric 27,46,139
  query 86-7
  unattributed 28
  variable 76,78,82,83,124,142,143
fields
  attributes 27,31
  creation of 25-37
  subfields 92,131
file types
  header 38,119,130,209,210
  key 52,53,69-70,130-1
  relative 38,119,122,129,130,135,137,209
  sequential 122,124,125-6,129,131,132,137,148,181-2,198-9

filename 36-7,39-40,125-6,127,160
files
  create 24-37,161
  rename 120-1,163
  scratch 120,164
formatting disks 15,21-2,121,156
header 72,80,144
insert/delete line 33,84,85
key (for C-64) 8
keyboard 13,42,202
labels 104-107,146,147,168-73
left-justification 45
loading programs 16-7,179
match count 95,151
menus 174-5
modifying records 65-70,182-4
out-of-field text 26,88,139,140
pagination 144
PaperClip interface 131
passwords 133-5,175-6
printer 4
  device number 71
  direct printer commands 142
  setup 71-72,128,144,177-9
printouts 82,72-3
  extra-wide 141
  forms 108-114,123,125,147,165-8
  header 72,80,144
  labels 104-107,146,147,168-73
  multi-page 143
  records 184-6
  screen 191
record format 25-37,125-6,138-40,160,161
record number 54,58-9,70
records
  add 42-52,180-2
  delete 67-8,183-4
  update 65-70,182-3
  view 54-64,186-7
relative files see file types
reports 74-87,124,138,140-1,144,187-91
right-justification 45
ROM Chip 8,201
scratch files 120,164
searches 54-64,65,79-80,85,139,192-6
  any match 54,62-3
  interrupt 82,191-2
  logical-and 147
  string 54,55-60
  wild card 54,61-2,64

security 133-5,204
  passwords 133-5,175-6
  secured file 134-5,196
  secured program, create 133-4,197-8
sequential files see file types
show all files 38,119,164
sorts 69,139-40,158,203
  special sorts 88-92
  subsorts 91-2,139-40,203
subfields 92,131
subfiles 147-8
support programs 130-1
  HDR-READ 130
  KEY-READ 130-1
  REL-READ 130,135
  SEQ-READ 131
  SUPERSORT 130,131,203
updating records 65-70,182-3
user conventions 11
view records 54-64,186-7
workdisk 21